Form Number A4606
Part Number D301055X012
February 2002

MODBUS PROTOCOL EMULATION PROGRAM
User Manual

Revision Tracking Sheet

February 2002

This manual may be revised from time to time to incorporate new or updated information. The revision level of each page is indicated at the bottom of the page opposite the page number. A major change in the content of the manual also changes the date that appears on the front cover. Listed below is the revision level of each page.

Page	Revision
Pages iii through vi	7/95
Pages 1-4, 1-5	2/02
Section 1 (all other)	7/98
Section 2 (all)	7/98
Section 3 (all)	7/98
Section 4 (all)	7/98
Section 5 (all)	7/95
Appendix A (all)	7/98
Appendix B (all)	2/96
Appendix C (all)	3/95
Appendix D (all)	5/97
Appendix E (all)	7/95

© Fisher Controls International, Inc. 1991-2002. All rights reserved.

Printed in the U.S.A.

While this information is presented in good faith and believed to be accurate, Fisher Controls does not guarantee satisfactory results from reliance upon such information. *Nothing contained herein is to be construed as a warranty or guarantee, express or implied, regarding the performance, merchantability, fitness or any other matter with respect to the products*, nor as a recommendation to use any product or process in conflict with any patent. Fisher Controls reserves the right, without notice, to alter or improve the designs or specifications of the products described herein.

Table of Contents

SECTION 1 — GETTING STARTED 1-1
1.1 USER MANUAL OvERVIEW 1-1
1.2 Scope of This Manual 1-1
1.3 Organization of This Manual 1-1
1.4 Modbus Protocol Emulation Program overview 1-2
1.4.1 Modbus Compatibility 1-2
1.4.2 Data Link 1-3
1.5 Versions and Program Names 1-4
1.6 Editor Keys 1-5

SECTION 2 — PROGRAM INSTALLATION 2-1
2.1 preparing to DOWNLOAD THE PROGRAM 2-1
2.2 Download procedure for GV101 version 1.4 2-1
2.2.1 Clear User Enable Flag 2-2
2.2.2 Download Utility 2-2
2.2.3 Initializing and Starting The Modbus Emulation Program 2-3
2.3 Download procedure for GV101 version 1.51 and later 2-4
2.4 procedure for downloading to flash memory 2-6

SECTION 3 — MODBUS CONFIGURATION 3-1
3.1 Configuration Access 3-1
3.2 Modbus Configuration parameters 3-2
3.3 Modbus Functions Configuration 3-5
3.4 Modbus Convert code descriptions 3-8
3.5 MODBUS HOST 3-14
3.5.1 Modbus Host Configuration 3-15
3.6 Modbus host Communication Parameters 3-17
3.7 Modbus Dial-up Operation 3-19
3.8 Modem Control 3-20
3.9 Controlling Modbus Host by FST 3-21

SECTION 4 — MODBUS MESSAGE FORMAT 4-1
4.1 MODES OF TRANSMISSION 4-1
4.2 Modbus MESSAGE CONTENTS 4-1
4.2.1 ASCII Message Framing 4-2

4.2.2 Remote Terminal Unit (RTU) Message Framing 4-2
4.3 EXPLANATION OF FUNCTIONS 4-3
4.3.1 Function Codes 01 and 02 - Read Output and Input Status 4-3
4.3.2 Function Codes 03 and 04 - Read Output and Input Registers 4-4
4.3.3 Function Code 05 - Force a Single Coil 4-5
4.3.4 Function Code 06 - Preset a Single Register 4-6
4.3.5 Function Code 15 - Force Multiple Coils 4-7
4.3.6 Function Code 16 - Preset Multiple Registers 4-8
4.3.7 Exception Response 4-9
4.4 special FUNCTIONS 4-10
4.4.1 Historical Data Storage 4-10
4.4.2 Address Table 4-13
4.4.3 Discrete Outputs 4-13

SECTION 5 — PROBLEM SOLVING 5-1
5.1 ROC Fails to Respond to Host Requests 5-1
5.2 ROC Error Messages 5-1
5.3 ROC Responds with Wrong Data 5-1
5.4 ROC Appends a Character to Message 5-1
5.5 ROC Does Not Respond in Point to Multi-Point Comm System 5-2
5.6 Deleting User-Defined Points 5-2
5.7 Integer-To-Float Conversion 5-2
5.8 ROC Response Preceded By Modem Commands 5-2
5.9 Deleting A User Program from ROC407 Flash Memory 5-3

APPENDIX A — ROC POINT TYPES A-1
A.1 ROC POINT AND DATA TYPES A-1
A.2 roc pOINT PARAMETER DEFINITIONS A-2

APPENDIX B — PROGRAMMING EXAMPLE B-1
B.1 Host ROC FUNCTION SEQUENCE TABLE #1 ..... B-1
B.2 Host ROC FUNCTION SEQUENCE TABLE #2 ..... B-6
B.3 Field ROC FUNCTION SEQUENCE TABLE #1 ..... B-22

APPENDIX C — MODEM REFERENCE C-1
C.1 Example Modem Commands C-1
C.2 RS232 Card to Dial-up Modem Cable C-4
C.3 Operator Interface port to Dial-up Modem Cable C-4

APPENDIX D — COMMUNICATION WIRING D-1

APPENDIX E — CONFIGURATION WORKSHEETS E-1

List of Figures and Tables

Figure 2-1. File Selection Display 2-4
Figure 2-2. File Download Display 2-5
Figure 3-1. Configure User Data Type Display 3-1
Figure 3-2. Modbus Configuration Display 3-2
Figure 3-3. Typical Modbus Functions Configuration Display 3-6
Figure 3-4. Modbus Host Configuration Display 3-15
Figure 3-5. Comm Port Configuration Display 3-17
Figure 3-6. Modem Control Display 3-20
Figure B-1. FST1 Registers Display B-5
Table 1-1. ROC-Supported Modbus Function Codes 1-3
Table 1-2. Editor Keys 1-6
Table 3-1. Modbus Function Convert Codes 3-9
Table 3-2. Status of Host Request or Command 3-16
Table 4-1. ASCII Message Format 4-2
Table 4-2. RTU Message Format 4-2
Table 4-3. Function Code 01 Host Request Example Message 4-3
Table 4-4. Function Code 01 ROC Response Example Message 4-4
Table 4-5. Function Code 03 Host Request Example Message 4-4
Table 4-6. Function Code 03 ROC Response Example Message 4-5
Table 4-7. Function Code 05 Host Request Example Message 4-5
Table 4-8. Function Code 05 ROC Response Example Message 4-6
Table 4-9. Function Code 06 Host Request Example Message 4-6
Table 4-10. Function Code 06 ROC Response Example Message 4-7
Table 4-11. Function Code 15 Host Request Example Message 4-7
Table 4-12. Function Code 15 ROC Response Example Message 4-8
Table 4-13. Function Code 16 Host Request Example Message 4-9
Table 4-14. Function Code 16 ROC Response Example Message 4-9
Table 4-15. ROC Exception Response Example Message 4-9
Table 4-16. Modbus Protocol Errors that Elicit an Exception Response Message 4-10
Table 4-17. History Data 4-11
Table 4-18. History Point Numbers for Database RAM 4-12
Table 4-19. Host Request for History Data Example Message 4-12
Table 4-20. ROC Response for History Data Example Message 4-13
Table A-1. Valid ROC Point Types A-1
Table A-2. Data Types A-2
Table A-3. Point Type 0, Configurable Opcode A-3
Table A-4. Point Type 1, Discrete Input Parameters A-4
Table A-5. Point Type 2, Discrete Output Parameters A-5
Table A-6. Point Type 3, Analog Input Parameters A-5
Table A-7. Point Type 4, Analog Output Parameters A-6
Table A-8. Point Type 5, Pulse Input Parameters A-6
Table A-9. Point Type 6, PID Parameters A-7

Table A-10. Point Type 7, AGA Flow Parameters A-8
Table A-11. Point Type 9, Local Display Panel (LDP) Parameters A-9
Table A-12. Point Type 10, AGA Flow Value Parameters A-10
Table A-13. Point Type 11, Tank Parameters A-11
Table A-14. Point Type 12, Clock Parameters A-11
Table A-15. Point Type 13, System Flag Parameters A-12
Table A-16. Point Type 14, Communication Port Parameters A-12
Table A-17. Point Type 15, System Variables A-13
Table A-18. Point Type 16, FST Register Parameters A-14
Table A-19. Point Type 17, Soft Point Parameters A-15
Table A-20. Point Type 19, Database Setup Parameters A-15
Table A-21. Point Type 20, ROC Task Parameters A-16
Table A-22. Point Type 40, MVS (205 Sensor) Parameters A-17
Table A-23. Point Type 41, AGA Run Parameters A-18
Table A-24. Point Type 42, Extra AGA Run Parameters A-20
Table A-25. Point Type 43, User List Parameters A-21
Table C-1. Modem S-Register Summary C-2
Table C-2. Dial Modifier Command Summary C-2
Table C-3. AT Command Set Summary C-3
Table C-4. Ampersand Command Summary C-3
Table C-5. Result Code or Status Messages Summary C-4
Table D-1. ROC300-Series Communications Signals D-1
Table D-2. ROC407 Communications Signals D-2
Table D-3. ROC-to-ROC Connections for Modem Cards (4-Wire RJ-11) D-2
Table D-4. ROC-to-ROC Connections for EIA-232 Serial Cards D-2
Table D-5. ROC-to-ROC Connections for EIA-422 Serial Cards D-3
Table D-6. ROC-to-ROC Connections for EIA-485 Serial Cards D-3

SECTION 1 — GETTING STARTED

1.1 USER MANUAL OVERVIEW

The Modbus Protocol Emulation Program is designed to allow the Remote Operations Controllers (ROCs) to emulate the communications protocol used by Modbus devices. This makes it possible to integrate the ROC and Modbus devices into the same host/slave system.

1.2 SCOPE OF THIS MANUAL

This manual describes how to configure and use the Modbus Protocol Emulation Program. The Modbus Protocol Emulation Program is configured by using either the ROCLINK or GV101 Configuration Software. The software uses an IBM compatible computer and the MS-DOS® operating system.

Menu descriptions in this manual are for version 1.5 (or later) of the GV101 software and version 1.70 or later of the Modbus Protocol Emulation program. **Note that this interim manual does not discuss how to use the ROCLINK Configuration Software (version 2.0 or later),** which is required for downloading and configuring the Modbus user programs for a ROC with a FlashPAC. Refer instead to the ROCLINK User Manual (Form A6050) dated February 1998 or later, or refer to a later edition of this Modbus manual (Form A4606).

1.3 ORGANIZATION OF THIS MANUAL

In this manual the sections are arranged to provide information in the order in which it is needed for first-time users. Once the user becomes familiar with the procedures, and the software is running in a ROC, the manual can be used as a reference tool.

The manual is organized into the following major sections:

Section 1	Getting Started
Section 2	Program Installation
Section 3	Modbus Configuration
Section 4	Modbus Message Format
Section 5	Problem Solving
Appendix A	ROC Point Types
Appendix B	Progamming Examples
Appendix C	Modem Reference
Appendix D	Communications Wiring
Appendix E	Configuration Worksheets

1.4 MODBUS PROTOCOL EMULATION PROGRAM OVERVIEW

The Modbus Protocol Emulation program allows a Remote Operations Controller (ROC) to emulate the Modbus Protocol. This makes it possible to integrate a ROC into a Modbus host system or to use the ROC as a Modbus system host in the Modbus host mode.

The Modbus Protocol Emulation program, when running on the ROC364, provides the Modbus Protocol at COM1, COM2, and the Operator Interface port. The Operator Interface port can use the protocol within the limitation that protocol switching is not hardware supported (using the DCD switch option) and modem dial-out is not supported. An external modem may be used when operating as a slave Modbus device.

The Modbus Protocol Emulation program, when running on the ROC306 or ROC312, provides the Modbus Protocol at the COMM port and the Operator Interface port. The limitations discussed for the ROC364 do not apply to the ROC306/312. The ROC306/312 supports both modem dial-out and port switching.

The Modbus Protocol Emulation program, when running on the ROC407 FloBoss, provides the Modbus Protocol at the COM1 port and COM2 port. No program is provided to run on the Operator Interface port. The ROC407 FloBoss also supports both modem dial-out and port switching.

The program is configured and downloaded to ROC RAM by the Type GV101 Configuration Software. Modbus functions are configured by point type and parameter. The ROC uses physical addressing of I/O, while Modbus applications use logical I/O addressing to retain compatibility with Modbus.

1.4.1 Modbus Compatibility

The Modbus Protocol Emulation program functions 1, 2, 3, 4, 5, 6, 15, and 16 use the same command and response format as the functions listed in the *Gould Modbus Protocol Reference Guide (January 1985)*, form PI-MBUS-300 Rev B. Depending upon the parameter configured (Byte, Word, Double Word, or Floating Point), functions 3, 4, and 6 can contain more data bytes than the other functions. The data byte upper limit is 240 bytes.

ROC point types and parameters for Modbus functions 1, 2, 3, 4, 5, 6, 15, and 16 are configured using the GV101 Configuration Software. Any supported ROC point type and parameter can be selected for function codes 3, 4, 6, and 16. Function codes 1, 2, 5, and 15 are restricted to input and output status point types. Table 1-1 provides details of the Modbus function codes supported by the ROC.

Table 1-1. ROC-Supported Modbus Function Codes

Code	Meaning	Action
01	Read Logic Coil Status	Obtain current status (ON/OFF) of a group of logic coils.
02	Read Discrete Input Status	Obtain current status (ON/OFF) of a group of discrete inputs.
03	Read Output Registers (Holding)	Obtain current binary value in one or more holding registers.
04	Read Input Registers	Obtain current binary value in one or more input registers.
05	Force Single Logic Coil	Force logic coil to a state of ON or OFF.
06	Preset Single Holding Register	Place a specific binary value into a holding register.
15	Force Multiple Logic Coils	Force a series of consecutive logic coils to defined ON or OFF states.
16	Preset Multiple Holding Registers	Place specific binary values into a series of consecutive holding registers.

1.4.2 Data Link

The data link between the host device and the ROC can be a serial data link operating at up to 9600 baud, or any of the following ROC communications cards:

- EIA-232 Serial Communications Card
- EIA-422/485 Serial Communications Card
- Radio Modem Communications Card
- Leased-Line Modem Communications Card
- Dial-Up Modem Communications Card

The data link can operate in the RTU (Remote Terminal Unit) mode or the ASCII (American Standard Code for Information Interchange) mode. In the RTU mode, data is sent in 8-bit binary characters. In the ASCII mode, each RTU character is divided into two 4-bit parts that are represented by their hexadecimal equivalent. The ASCII mode uses twice as many characters as the RTU mode. Each character sent is composed of a Start bit, 8 or 7 Data bits, and one or two Stop bits with Even, Odd, or No parity. The communications parameters are defined in the ROC Comm Ports configuration screen of the GV101 configuration software.

1.5 VERSIONS AND PROGRAM NAMES

All ROCPAC and FlashPAC versions for the ROC306/ROC312, and ROCPAC/FlashPAC versions greater than 1.50 for the ROC364, and all versions of the ROC407 FloBoss support program memory allocations. Version 1.40 (and greater) of the GV101 Configuration Software supports the downloading and viewing of programs that are compiled for variable memory allocation to the ROC306/ROC312 and the ROC364. Version 1.61 (and greater) of the GV101 Configuration Software supports the downloading and viewing of programs that are compiled for variable memory allocation to the ROC407 FloBoss. ROCLINK Configuration Software supports downloading and viewing of all variable memory allocation programs.

If you are not using a memory allocation version, you must use the MB1B0A0 and MB2B8A8 programs. MB1B0A0 and MB2B8A8 programs require RAM in Memory Expansion Slot 3, available in the ROC364 only, or in a 256K RAM module.

The other Modbus programs are named by the location in memory in which they reside. The current names for Modbus programs are, along with other pertinent information:

File	ROC Type	Port	Code	Data
MB0D0D8.H00	ROC300 ROCPAC	LOI	D0000-D7FFF	D8000-DBFFF
MB16070.H00	ROC300 ROCPAC	Comm 1	60000-67FFF	70000-73FFF
MB1B0A0.H00	ROC300 ROCPAC	Comm 1	B0000-B7FFF	A0000-A3FFF
MB1C0C8.H00	ROC300 ROCPAC	Comm 1	C0000-C7FFF	C8000-CBFFF
MB2B8A8.H00	ROC300 ROCPAC	Comm 2	B8000-BFFFF	A8000-ABFFF
MB2D0D8.H00	ROC300 ROCPAC	Comm 2	D0000-D7FFF	D8000-DBFFF
FPMB0_C0.H00	ROC300 FlashPAC	LOI	C0000-C7FFF	A4000-A7FFF
FPMB1_C8.H00	ROC300 FlashPAC	Comm 1	C8000-CFFFF	AC000-AFFFF
FPMB2_C0.H00	ROC300 FlashPAC	Comm 2	C0000-C7FFF	A4000-A7FFF
FPMB1_D8.H00	ROC300 FlashPAC	Comm 1	D8000-DFFFF	B4000-B7FFF
FBMB1_60.H00	ROC407 FloBoss	Comm 1	60000-67FFF	68000-6BFFF
FBMB1_A0.H00	ROC407 FloBoss	Comm 1	A0000-A7FFF	6C000-6FFFF
FBMB1_05.H00	ROC407 (v1.05)*	Comm 1	A0000-A7FFF	6C000-6FFFF
FBMB2_70.H00	ROC407 FloBoss	Comm 2	70000-77FFF	78000-7BFFF
FBMB2_A8.H00	ROC407 FloBoss	Comm 2	A8000-AFFFF	7C000-7FFFF
FBMB2_05.H00	ROC407 (v1.05)*	Comm 2	A8000-AFFFF	7C000-7FFFF
FB2_0570.H00	ROC407 (V1.05)*	Comm 2	70000-77FFF	78000-7BFFF
FBCLR_A0.H00	ROC407 FloBoss		A0000-A3FFF	
FBCLR_A8.H00	ROC407 FloBoss		A8000-ABFFF	

*This program file is intended for a FloBoss 407 with firmware version 1.05 or greater. The file will load into earlier firmware versions, but will not run.

If you're using ROCLINK Configuration Software, refer to the User Programs procedure, in either the *ROCLINK User Manual (for DOS)*(Form A6051) or *ROCLINK for Windows User Manual* (Form A6091).

If you're using the GV101 Configuration Software, to view the memory allocation in a ROC, select Utilities. From the ROC Utilities Menu, select User Program Routines. Then select the User Program Routines, and then the Check User Memory Allocation option. A display appears that shows the allocation status of all user memory blocks in the upper half of the screen, and a list of the user programs and task status in the lower half. Memory blocks are 16K bytes in length and their allocation is determined by the individual user program. The ROC Operating System checks each block for proper allocation and does not permit the allocation of two different programs to the same block.

The upper half of the display appears similar to the following:

```
Block 6000 = Unalloc   Block a000 = Unalloc   Block c000 = Code 1
Block 6400 = Unalloc   Block a400 = Unalloc   Block c400 = Code 1
Block 6800 = Unalloc   Block a800 = Unalloc   Block c800 = Data 1
Block 6c00 = Unalloc   Block ac00 = Unalloc   Block cc00 = Unalloc
Block 7000 = Unalloc   Block b000 = Unalloc   Block d000 = Code 2
Block 7400 = Unalloc   Block b400 = Unalloc   Block d400 = Code 2
Block 7800 = Unalloc   Block b800 = Unalloc   Block d800 = Data 2
Block 7c00 = Unalloc   Block bc00 = Unalloc   Block dc00 = Unalloc
```

For the ROC306/312 and ROC364, blocks 6000 through bc00 correspond to various RAM modules, and blocks c000 through dc00 correspond to the RAM in a ROCPAC module. For the ROC407 FloBoss, blocks 6000 through 7c00 correspond to user RAM in the ROC, blocks a000 through cc00 correspond to flash memory in the ROC, and blocks d000 through dc00 are reserved for factory use.

The allocation status since the last Warm Start is shown for each memory block as follows:

`No RAM` -- No RAM is installed in this location (GV101 Rev 1.5 or later). When viewing ROC407 memory, "No RAM" displayed for flash memory blocks a000 through cc00 means that the memory is unallocated.
`Unalloc` -- No program assigned or no memory installed.
`Code` *x* -- Code location for user program *x*.
`Data` *x* -- Data location for user program *x*.

1.6 EDITOR KEYS

The softkeys described in Table 1-2 are located along the bottom of the display and are used to operate the GV101 Configuration software and manipulate the Modbus protocol configuration displays. All keys except for cursor position and ESC are softkeys whose labels appear in the displays.

Table 1-2. Editor Keys

Key	Description
Update	Update the information in the display.
Prev., Next	Move between the different function codes.
RD Disk	Read data from a disk file. A prompt appears requesting the drive and directory to be searched for *.scg files. After selection of the directory, all files with the .scg extension are listed. Select the file to read by following the instructions on the display. If the file does not match the current point type being configured, an error "Data Type Mismatch" is displayed and the read file is aborted.
WR Disk	Save the data for a configured function. A prompt appears requesting the name of the file to save. Use a name that is descriptive of the function to be saved. The suggested convention is: R1C13A .scg, where: R1 is the ROC number (address 1) C1 is the communications port number (port 1) 3A is the function number .scg is the extension (Do not enter)
Save	Save the configuration after it has been edited. After the SAVE key has been pressed, a prompt appears asking if the data was meant to be saved. If yes, press ENTER and the data is sent to the ROC and logged in the Event Log.
ESC, Quit	Quit the Function Configuration Mode and return to the Configure Menu.
↑, ↓	Move between data fields within a function.
←, →	Move between characters within a data field.

SECTION 2 — PROGRAM INSTALLATION

This section provides instructions for installing the Modbus Protocol Emulation Program into ROC memory. Make sure to read Section 1.5 of this manual for program and memory requirements.

If you're using the ROCLINK Configuration Software for program installation, refer to Section 8.2, User Programs, in the *ROCLINK User Manual (Form A6051).*

2.1 PREPARING TO DOWNLOAD THE PROGRAM

To install the Modbus Protocol Emulation Program, connect an IBM-compatible computer containing the GV101 configuration software to the ROC Operator Interface port. Before the downloading process is started, make sure the RAM is available in the ROC for the intended download.

To run the GV101 software, perform the following steps:

Ensure that the current disk drive and directory is the one in which the GV101 software is installed.

> Note: If you have added the drive and directory of the GV101 software to the PATH statement in your AUTOEXEC.BAT file, then you can skip this step. Alternately, if a batch file (named GV101.BAT) has been written that sets the drive and directory and is locatable by the PATH statement, you can skip this step. See your DOS user manual for help on modifying the PATH statement or writing a batch file.

To change the current drive, at the DOS prompt type in the drive letter followed by a colon (such as `C:`) and press ENTER. To change the current directory, type in CD followed by a backslash and the directory name (such as `CD \GV101`), and then press ENTER.

Next, type in GV101 at the DOS prompt, and press ENTER.

The GV101 Configuration Software then loads and initializes. This may take from 1 to 5 seconds, depending on the speed of your computer.

When the GV101 Software is done loading, log on by entering your identification and code number.

2.2 DOWNLOAD PROCEDURE FOR GV101 VERSION 1.4

The download procedure in this section is used with GV101 Configuration Software version 1.4. If you have a more recent version, refer to Section 2.3 or 2.4 as appropriate.

2.2.1 Clear User Enable Flag

If you are using a ROC364 with a ROCPAC version 1.50 or earlier, you need to check the User Com1 and Com2 Enable flags before downloading Modbus Protocol Emulation Program. Using the GV101 Configuration Software, select the Configure menu, and then the ROC System Flags option. In the screen that appears, User Com1 Enable and User Com2 Enable flags must be set to "0" to disable operation of any user programs that may be running tasks on COMM1 or COMM2. Otherwise, the ROC Operating System will detect an error and enable the Watchdog Timeout. Later ROC and GV101 versions perform this step automatically.

2.2.2 Download Utility

To download the Modbus Protocol Emulation Program, select the Utilities option from the Main Menu of the GV101 Configuration Software. From the ROC Utilities Menu, select the User Program Routines option. In the resultant User Routines menu, select the Download User Programs option to download to RAM. This option checks the operating system to see if any user application programs are currently enabled or running. If one or more programs are running, a prompt appears requesting permission to disable the programs by clearing the User Enable flags and performing a Warm Start. Pressing ENTER disables the User Enable flags, and then proceeds with the rest of the download procedure as described below. Pressing ESC cancels the download procedure and returns you to the User Routines menu.

When prompted during the download procedure, specify the drive and directory that contains the Modbus program files and press ENTER. The GV101 software then displays all files in the specified drive and directory with the .H00 extension. Select the desired Modbus program by entering the corresponding number.

After you verify your selection, the downloading process is started and information similar to the following is displayed:

```
Downloading File a:MB1C0C8.H00

        Data Sent nnnnn
          Segment nnnn
           Offset nnnn
            Bytes nnn
```

When the file has been downloaded to RAM, the following message is displayed:

```
Download Complete, Press <ENTER> to Configure Flags
 <Esc> to Quit, <Enter> to continue.
```

Press ESC to return to the User Routines menu without running the Modbus program. Pressing ENTER automatically brings up the ROC Flags display that allows you to set the needed User Enable flags for the Modbus program.

2.2.3 Initializing and Starting The Modbus Emulation Program

Before the Modbus Protocol Emulation Program (as well as any previously loaded user programs) can be run, the User Enable flag associated with the loaded program must be set. Enable the Modbus program by setting the User Enable flag (User OP Port Enable, User Com1 Enable, or User Com2 Enable) to "1". If there are other programs that you want to run, likewise enable the flags for them. Remember to perform a Save operation to transmit the value to the ROC.

To begin execution of the Modbus program, as well as any other user programs loaded into the ROC, one of the following restart sequences must be initiated:

Warm Start -- Initiated when the Warm Start flag (in the ROC Flags display) is set to "1" and saved. This method has the least effect on the ROC. All I/O scanning is temporarily suspended, but the I/O retains its last state. History, events, and alarms are not affected.

Cycling Power -- Initiated by turning the ROC power off and then back on. **Do not use this method if the ROC has active analog or discrete outputs. Cycling power causes power to be removed from the control devices.**

Cold Hard Start -- Initiated when the Cold Hard Start flag (in the ROC Flags display) is set to "1" and saved. **This method causes all configuration data to be reloaded from EEPROM or from ROM. All history, events, and alarms are cleared.** This method will not work if the User Enable flags have not been previously saved to EEPROM, because the old values in EEPROM are used.

In most cases, the Warm Start is the preferred method to get the user programs running.

2.3 DOWNLOAD PROCEDURE FOR GV101 VERSION 1.51 AND LATER

The download procedure in this section is used with GV101 Configuration Software version 1.51 and later. However, if you are downloading the Modbus program to flash memory in a ROC407, use the procedure given in Section 2.4.

To download the Modbus Protocol Emulation Program, select the Utilities option from the Main Menu of the GV101 Configuration Software, and then select the User Program Routines option. Next, select the Download User Programs option. This utility allows you to load one or more user programs from a disk file into the ROC user memory. Use the Check User Memory Allocation routine (discussed in Section 1.5) if you want to see what user programs are currently loaded. Since the Modbus program has files that can be loaded selectively into various memory segments, this may help you decide which program files to use.

When you start the utility, a display appears for selecting the program files to be loaded. This display (see Figure 2-1) lists the names of all the files that have the .H00 extension and are located in the default drive and directory.

Use the "up" and "down" arrow keys to indicate the desired file in the list and then press ENTER. If the desired file is not listed, you can instead select a drive and directory that has the file you want. Note that the current drive and directory are given at the top of the list.

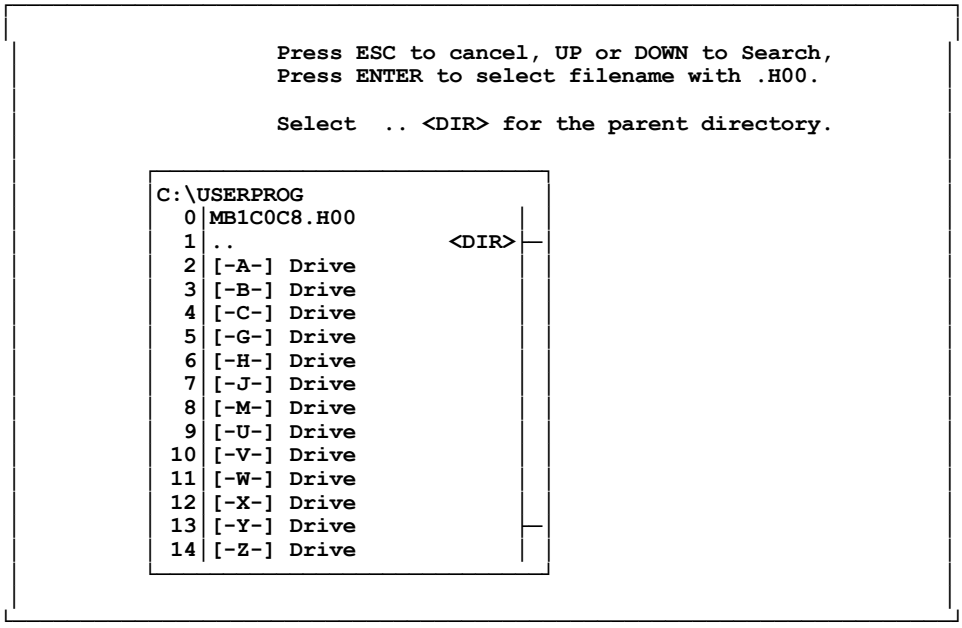

Figure 2-1. File Selection Display

After you select a valid file and press ENTER, the following information is displayed:

```
File you selected is A:\MB1C0C8.H00

                        Is This Correct ?
Esc = cancel   F1 = Try again      F2 = OK
```

Pressing ESC returns you to the User Routines menu, pressing F1 allows you to select a different disk file, and pressing F2 brings up a new display, as shown in Figure 2-2.

Press ESC to exit the download procedure and return to the User Routines menu. If there are other program files you want to download at this time, press F1. When you press F1, you are returned to the file selection display (shown in Figure 2-1), from which you can select and verify another program file. You can repeat the process of adding more files to be downloaded, up to a maximum of eight files.

If there is a file listed in the file download display that you decide you don't want to download after all, you can remove it from the list by pressing "D", typing in the number of the file to remove, and pressing ENTER. The file is removed from the list, and the list is renumbered.

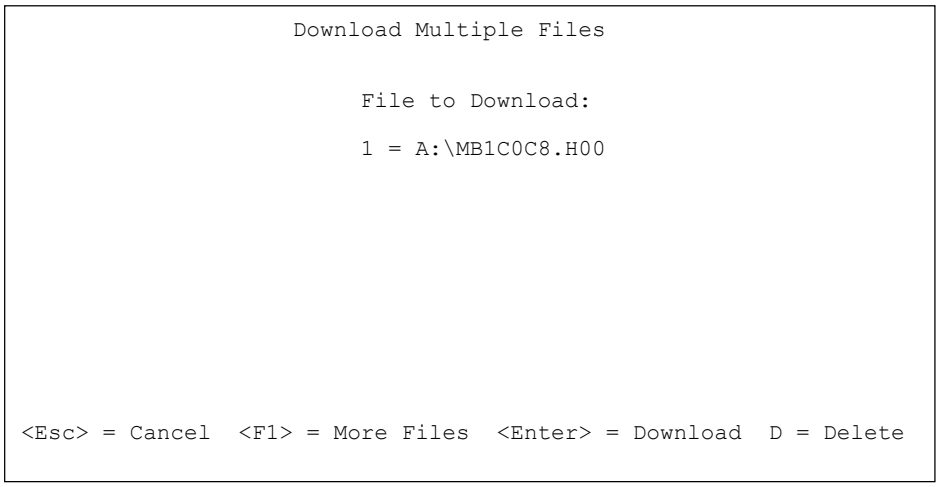

Figure 2-2. File Download Display

When you are satisfied with the download list, press ENTER. You are presented with the following options:

```
1 = Download file(s)
2 = Cold Hard Start and Download file(s)
3 = Exit Back to File Selection Menu
```

Choose options 1 or 2 for downloading the program files, or option 3 for returning to the download list by typing in the number and pressing ENTER. **Note that option 2 performs a cold hard start, which reloads all configuration data from EEPROM, clears all history, events, and alarm logs, and disables all user program tasks and User Data Types.** A caution to this effect appears on the screen

if you select option 2; press ENTER again if you are sure you want to continue. A cold hard start can take up to 30 seconds to be completed.

When downloading is initiated, the software first checks to see if there are any user programs in the targeted memory. If so, it automatically clears the appropriate User Enable flags, performs a Warm Start, and **clears the existing user programs from the targeted memory.** The software then begins the downloading process. Pressing ESC anytime during downloading cancels the process and returns you to the User Routines Menu.

During the file download (which can take up to several minutes for each file), information similar to the following is displayed:

```
Downloading File A:\MB1C0C8.H00

          Data Sent nnnnn
            Segment nnnn
             Offset nnnn
              Bytes nnn
```

When the file or files have been downloaded, the software displays the following message:

```
Download complete, Press <ENTER> to Enable Flags Automatically
<Esc> to Cancel.
```

Press ENTER to automatically enable the appropriate User Enable flags with a "1" and perform a Warm Start. Note that certain user programs can be enabled with values other than a "1". If you have just loaded such a program (refer to its user manual for more information) and wish to set the User Enable flag to a "2" or "3", or if for some reason you don't want to start running the user programs yet, then press ESC to bypass automatic enabling of the User Enable flags. If you bypass automatic enabling, you will need to use the ROC Flags display to set the proper User Enable flags and perform a Warm Start.

If you chose to enable User Flags automatically, the following appears on the screen:

```
Wait for Warm Start.......

Flag Enable Complete, press any key to Continue
```

You are then returned to the User Routines menu.

2.4 PROCEDURE FOR DOWNLOADING TO FLASH MEMORY

If you're using the ROCLINK Configuration Software for program installation, refer to Section 8.2, User Programs, in the *ROCLINK User Manual (Form A6051).* ROCLINK is required for downloading to the FlashPAC in a ROC300-Series unit.

The download procedure in this section is used only to download the Modbus program to the flash memory in a ROC407. This procedure requires using GV101 Configuration Software version 1.61 (or later). If you are downloading the Modbus program to RAM in a ROC407, use the procedure given in Section 2.3.

To download the Modus Protocol Emulation Program to flash memory, select the Utilities option from the Main Menu of the GV101 Configuration Software, and then select the User Program Routines option. In the resultant User Routines menu, select the Program Flash Memory option. This utility allows you to load a program from a disk file into the ROC407 flash memory.

Use the Check User Memory Allocation routine (discussed in Section 1.5) if you want to see whether or not a user program is currently loaded in the intended memory area. Since the Modbus program has at least two files that can be loaded into different segments of flash memory, this may help determine which file to use. Note that the downloading process for flash memory does not clear an already loaded program out of the targeted memory area. If you need to clear a program from flash memory to make room for the Modbus program, refer to Section 5.9.

When you start the Program Flash Memory utility, a display appears for selecting the program files to be loaded. This display (see Figure 2-1) lists the names of all the files that have the .H00 extension and are located in the default drive and directory. Use the "up" and "down" arrow keys to indicate the desired file in the list and then press ENTER. If the desired file is not listed, you can instead select a drive and directory that has the file you want. Note that the current drive and directory are given at the top of the list.

After you select a valid file and press ENTER, the following information is displayed:

```
File you selected is A:\FBMB1_A0.H00

                        Is This Correct ?
Esc = cancel   F1 = Try again      F2 = OK
```

Pressing ESC returns you to the User Routines menu, pressing F1 allows you to select a different disk file, and pressing F2 starts the download process and brings up the following message:

```
Downloading File A:\FBMB1_A0.H00

Waiting for setting Up Flash memory...
```

After displaying the above message for a few seconds, the following downloading message appears.

```
Downloading File A:\FBMB1_A0.H00

          Data Sent nnnnn
            Segment nnnn
             Offset nnnn
              Bytes nnn
```

Pressing ESC anytime during downloading (which can take several minutes) cancels the process and returns you to the User Routines Menu.

When the file has been downloaded, the software displays the following message:

```
Programming the Flash memory, please wait....
```

This message automatically clears after flash memory has been successfully programmed. You are then returned to the User Routines menu.

Before the Modbus Protocol Emulation Program (as well as any previously loaded user programs) can be run, the User Enable flag associated with the loaded program must be set. (The download routine for flash memory does not change the status of User Enable flags.) Make sure the Modbus program will run by setting the associated User Enable flag in the ROC Flags display (User Com1 Enable, or User Com2 Enable) to "1". If there are other programs that you want to run, likewise ensure that the associated flags are enabled for them.

To begin execution of the Modbus program, as well as any other user programs loaded into the ROC, a Warm Start should be performed. To do this, set the Warm Start flag (in the ROC Flags display) to "1". All I/O scanning is temporarily suspended during the Warm Start, but the I/O retains its last state. Remember to perform a Save operation to transmit the changed flag values to the ROC407.

SECTION 3 — MODBUS CONFIGURATION

The Modbus Protocol Emulation Program functions are configured using the GV101 or ROCLINK configuration software. The Modbus configuration information resides in the Modbus user program that is downloaded to the ROC. User-Defined Points (UDP) are used to make this data available to the configuration software. The Modbus program for COM1 port reserves UDP 32, UDP 34, UDP 35, and UDP 36 for configuration data. The program for COM2 and the Operator Interface port reserves UDP 33, UDP 37, UDP 38, and UDP 39 for configuration data. **Do not run Modbus on the Operator Interface port and COM2 at the same time in the ROC364.**

If you're using the ROCLINK Configuration Software to configure Modbus in a ROC300-Series or ROC407 unit, select the appropriate Modbus operation (such as Modbus Config COM1) in the User Data item under the Data menu (see Section 5.11 of the *ROCLINK User Manual)* and configure parameters similar to those described in the rest of Section 3 below.

3.1 CONFIGURATION ACCESS

NOTE: The computer running the GV101 Configuration Software must be connected to the ROC Operator Interface port before the configuration process is started.

From the Main Menu of the GV101 Configuration Software, select the Configure option. Then from the Configuration Menu, select the Configure User Data Types option. Figure 3-1 shows the User Data Types available for Modbus configuration.

```
Configure User Data Types

1  =  Modbus Modem COM1
2  =  Modbus Modem COM2
3  =  Modbus Modem OpPort*
4  =  Modbus Config COM1
5  =  Modbus Funct COM1
6  =  Modbus Host COM1
7  =  Modbus Config COM2
8  =  Modbus Funct COM2
9  =  Modbus Host COM2
10 =  Modbus Config OpPort*
11 =  Modbus Funct OpPort*
12 =  Modbus Host OpPort*

Select Type to Configure =
```

*ROC300-Series only.

Figure 3-1. Configure User Data Type Display

The "Modbus Config" selections allow the user to edit the Modbus configuration parameters. The "Modbus Funct" selections allow the user to edit the parameters of Modbus Function Codes 1, 2, 3, 4, 5, 6, 15, and 16. The "Modbus Host" selections allow the user to set up the Modbus configuration parameters to allow the ROC to act as a host. The "Modbus Modem" selections allow the user to edit the Modbus Modem control parameters. On selection of one of the above menu items, the GV101 Configuration Software enters the Editor Mode, as described in the rest of Section 3.

3.2 MODBUS CONFIGURATION PARAMETERS

The Modbus configuration parameters are set and edited with the Modbus Configuration display. Upon selection of a User-Data Type, the GV101 Configuration Software displays the configuration parameters as shown in Figure 3-2.

```
                        Modbus Config COM1   1 of 1

       ASCII = 0,  RTU = 1: 1                HI Float Scale 2 .0000000
     Byte Order 1 = MSB 1st 0                LO Float Scale 2 .0000000
           Host Enable = 1: 0                HI Float Scale 3 .0000000
         Log Data 1 = Yes: 0                 LO Float Scale 3 .0000000
          Init Memory = 1: 0                 HI Float Scale 4 .0000000
       Port Switch En = 1: 0                 LO Float Scale 4 .0000000
         DCD=0,DI=1,SPT=2: 0                 HI Float Scale 5 .0000000
          Modbus Baud Rate 1200              LO Float Scale 5 .0000000
          Switch Baud Rate 9600              HI Float Scale 6 .0000000
          Input Data Start 100               LO Float Scale 6 .0000000
         Output Data Start 300               HI Float Scale 7 .0000000
          HI Integer Scale 4095              LO Float Scale 7 .0000000
         LOW Integer Scale 0                 HI Float Scale 8 .0000000
            HI Float Scale 1 .0000000        LO Float Scale 8 .0000000
            LO Float Scale 1 .0000000



1         2 Prev.    3 Next    4 RD Disk    5 WR Disk  6 Quit   7    8 Save
Update
```

Figure 3-2. Modbus Configuration Display

The following paragraphs describe the Modbus configuration parameters as shown on the display.

ASCII = 0, RTU = 1 -- This parameter sets the operating format of the Modbus Protocol installed on the ROC. If set to "1", Modbus operates in the RTU Mode with CRC-16 Error Checking. If set to "0", Modbus operates in the ASCII mode with LRC Error Checking.

Byte Order 1=MSB 1st -- The order of data bytes in a transmission or request can be reversed by configuring the Byte Order parameter. The default value of "0" places the LSB first. MSB first is selected by setting this parameter to "1". This is only effective on the Data field of a Modbus message. It has no effect on the data for Function Codes 01, 02, and 05, which contain byte-only data.

Host Enable = 1 -- When set to "1", activates the Modbus host mode. Setting this parameter to "1" does not initiate a transmission, but enables the program to monitor the Mode field of the Comm Port Parameters. If bit 7 in the Mode Field of the Comm Port Parameters is set to "1", and Host Enable is "1", a transmission is initiated.

Log Data 1 = Yes -- When set to "1", allows any changes to parameters of the ROC to also be logged in the Event Log. When set to "0", the parameters are changed but not logged into the Event Log.

Init Memory = 1 -- When set to "1" and transmitted to the ROC, the Modbus program sets up internal registers and then resets this field to "0". On the next power-up, warm start, or cold hard start, the Modbus program initializes the memory associated with the program to default values.

CAUTION

Set this field to "1" only if a problem with Modbus memory is suspected. All Modbus configuration data will be lost for this communications port.

Port Switch En = 1 -- Enables the Modbus program to switch back and forth between ROC protocol and Modbus protocol dependent upon an external event. The switching enables one communications port to use both protocols. When this field is set to "0" and the Modbus program is enabled (see section 2.2 to enable Modbus), Modbus protocol is used on the port and protocol switching is ignored. When set to a "1", the condition specified by the DCD=0,DI=1,SPT=2 field is used to determine which protocol is to be active.

NOTE

The preceding parameter is supported by ROC306/ROC312 ROCPAC versions P1.00 or greater and ROC364 ROCPAC versions greater than P1.50.

DCD=0,DI=1,SPT=2 -- Only active when the "Port Switch En = 1" field is set to "1". It specifies the condition that will switch protocols. The possible conditions are:

0 = The Comm port signal is used to switch protocols. When active (1), the Modbus protocol is used. Uses DCD (pin 1) on Operator Port (ROC306/312 only; not suported on ROC364), and DSR (pin 6) on Comm 1 and Comm 2. These signals can be hardwired active by jumping them to the DTR (pin 4) signal. Normally used when cables are physically changed to activate switchover.

1 = The first Discrete Input is used to switch protocols. When inactive (0), the Modbus Protocol is used. Normally used when an external event triggers a need to switch protocols.

2 = The Integer Flag in the first Soft Point is used to switch protocols. When the value in the Integer Flag is zero, the Modbus protocol is used. Normally used when control of the port is determined by an FST application program.

Modbus Baud Rate -- The baud rate entered in this field is only used for the Modbus protocol when the "Port Switch En = 1" field is set to "1". If the Port Switch En field is set to "0", the baud rate is determined by the Communication parameters. Only used when a port switch is enabled.

Switch Baud Rate -- The baud rate entered in this field is only used for the ROC protocol when the "Port Switch En = 1" field is set to "1". If the "Port Switch En = 1" field is set to "0", the baud rate is determined by the Communication parameters. Only used when a port switch is enabled.

Input Data Start, Output Data Start, HI Integer Scale, LOW Integer Scale -- These four parameters make up the Modbus Address Table.

Some Modbus hosts might request from the ROC the Address Table for configuration of the host. Normally, during initialization, the host reads the Address Table to determine where the Input and Output data tables are located, and to determine the 100% and 0% values for all analog data. The Address Table normally resides at the start of the Modbus Registers, either at location 0 or 40000. The Address Table is assigned a register location by specifying a starting and ending address for Data Type 44 in Function Code 3 configuration.

The Address Table consists of four words (or registers) described as follows.

INPUT DATA START -- Contains the starting register address of the Input data table.
OUTPUT DATA START -- Contains the starting register address of the Output data table.
HI INTEGER SCALE -- Contains the 100% value for all analog data (Type 3, Parameter 17 and Type 4, Parameter 9) registers.
LOW INTEGER SCALE -- Contains the 0% value for all analog data (Type 3, Parameter 17 and Type 4, Parameter 9) registers.

In the ROC, the endpoints of the Analog Inputs and the Analog Outputs are used to scale or calibrate the range of the input or output signal. Since each I/O point can have different scaling, the Raw Values from the Analog I/O points are normalized to the values defined by the HI INTEGER SCALE and LOW INTEGER SCALE fields.

These values are signed integers, so they can range from zero to 32767. The default signal range is 800 to 4000, which represents a 1 to 5 volt or 4 to 20 milliamp signal. These data fields can also be used to scale the Analog I/O to integer values with implied decimal point.

For example, all Analog I/O Raw values can be transmitted with 0 to 1000 values (0 to 100.0, decimal point implied) by setting the values in this field to 0 for the Low Scale and 1000 for the High Scale. The scaling is used only on Analog I/O specified by I/O type 3 (AIN), parameter 17 (Raw A/D Input), and I/O type 4 (AO), parameter 9 (Raw D/A Output).

The HI INTEGER SCALE and LOW INTEGER fields are normally used when the host is not able to process floating point numbers.

HI Float Scale, LO Float Scale -- In host systems that do not accept floating point numbers, eight sets of floating point ranges for values can be specified. This allows floating point values, such as PID setpoints, to be read and set by the host as integer values. The ROCs floating point values are converted to integers by configuring a register or range of registers with the CONV field set in the Function Code configuration. The equations used to convert floating point to integer to enable the reading of floating point values are:

Float Range = Hi Float Scale - Low Float Scale
Integer Range = Hi Integer Scale - Low Integer Scale
Adj. Reading = Float Reading - Low Float Scale

$$\text{Integer} = \frac{\text{Integer Range} \times \text{Adj. Reading}}{\text{Float Range}} + \text{Low Integer Scale}$$

The equations used to convert integers to floating point values are:

Float Range = Hi Float Scale - Low Float Scale
Integer Range = Hi Integer Scale - Low Integer Scale
Adj. Integer = Integer Sent - Low Integer Scale

$$\text{Float Value} = \frac{\text{Adj. Integer} \times \text{Float Range}}{\text{Integer Range}} + \text{Low Float Scale}$$

The Hi/Lo float scale values are also used as high and low deadband limits when Convert Code 49 through 56 is selected (see Modbus Functions Configuration, CONV Field).

3.3 MODBUS FUNCTIONS CONFIGURATION

The Modbus function parameters are set and edited with the Modbus function parameters display in the GV101 software configuration program. Upon selection of a User Defined Function from the menu (Figure 3-1), the GV101 Configuration Software reads the Modbus Data Template and then displays the function parameters as shown in Figure 3-3. For configuration, the table template would display "0" for the values.

The function configuration tables are used to associate Modbus register numbers with ROC point data. When a Modbus request is received, the Modbus user program searches the function configuration table for the Modbus function requested, starting with the first table entry down to the last. If a register

```
                        Modbus Funct COM1  7 of 24

                                TAG ID FUNCT 3A

1  Start 10   End Addr 13   Type 44   Lgl# 0     Param 0    Conv 0
2  Start 100  End Addr 106  Type 3    Lgl# 0     Param 17   Conv 0
3  Start 107  End Addr 107  Type 10   Lgl# 0     Param 3    Conv 1
4  Start 108  End Addr 108  Type 10   Lgl# 1     Param 3    Conv 2
5  Start 109  End Addr 114  Type 16   Lgl# 128   Param 2    Conv 1
6  Start 115  End Addr 115  Type 6    Lgl# 0     Param 6    Conv 3
7  Start 116  End Addr 116  Type 6    Lgl# 0     Param 11   Conv 1
8  Start 117  End Addr 117  Type 11   Lgl# 0     Param 8    Conv 5
9  Start 118  End Addr 118  Type 11   Lgl# 0     Param 12   Conv 1
10 Start 119  End Addr 120  Type 11   Lgl# 128   Param 13   Conv 5



1 Update  2 Prev.  3 Next  4 RD Disk  5 WR Disk  6 Quit  7    8 Save
```

Figure 3-3. Typical Modbus Functions Configuration Display

number match is found, it builds a response based on the ROC point type and parameter configured in the table. If no register number match is located, an error message is returned. The user program will locate a register as long as it matches the beginning register number, the ending register number, or any number in between for that particular entry in the table.

When configuring the table, register numbers should be unique. Registers may be duplicated as long as they are located in a separate Modbus function configuration table. If a register number is duplicated within the same Modbus function table, the first occurrence will be used. Also it is best to number the table from smallest register number to largest especially when using two table entries to configure a continuous group of registers.

Up to thirty different requests can be configured for Modbus Functions 3, 4, and 16 (three displays of ten register ranges). By making the registers continuous, meaning the start address of a new line is one greater than the end address of the previous line, a continuous data table can be created up to the limit of 240 bytes. This type of data table would allow access to all data with one request.

Up to ten different requests can be configured for Modbus Functions 1, 2, 5, 6, and 15. For functions 1, 2, 5, and 15, the parameter specified should be a single-byte parameter type, preferably a status parameter (only bit 0 is used), since this function packs the data into a binary format for transmission.

Each address span must be unique within the function for proper operation. If not, the first valid address is used. A "0" in the type field disables that Request Number.

The following paragraphs describe the fields used in the Modbus Function parameters configuration display. There is a Modbus Function Configuration worksheet in Appendix E of this manual.

Start Field -- The Start Address is a number that represents the Data Register in a Modbus device. This number is used in the Value Offset field for a Modbus request. In certain Modbus host devices, the register 40101 is actually transmitted as "100". The value "100" should be placed in the start address field since the ROC used the actual number sent by the host. A Start Address of "0" is a valid address.
Any number from 0 to 65535 is valid. Register numbers can be duplicated as long as they are in separate Modbus Function configuration tables. The tables should be numbered from smallest to largest.

End Addr Field -- The End Address is a number that represents the last location of a point's data. The value for this number is computed by:

End Address = (Start Address + Number of Functions) - 1

Type Field -- The Type field denotes the type of data associated with an address. If the Type field is set to "0", the line in the configuration table is considered invalid and will be skipped when processing a request. The valid point types for the ROC are listed in Table A-1 in Appendix A. Appendix A has a complete list of parameters for each of the ROC Point Types.

Lgl# Field -- The logical number field specifies the logical number to be assigned to the start address. If the range of addresses is greater than "0", the next address (start address + 1) corresponds to the value specified by logical number + 1. This is called "horizontal indexing" of structures to specify data. For example, the configuration of:

Start 100 End Addr 106 Type 3 Lgl# 0 Param 17 Conv 0

specifies seven values for parameter 17 of Analog Inputs starting at the first Analog Input (LGL # 0). So register 100 = parameter 17 of AI #1, register 101 = parameter 17 of AI #2, ..., and register 106 = parameter 17 of AI #7.

Another example of horizontal indexing uses the configuration of:

Start: 120 End Addr: 120 Type: 17 Lgl#: 14 Param: 5 Conv: 0

specifies one value for parameter 5 of a Soft Point starting at the fifteenth Soft Point (Lgl# 14). So register 120 = parameter 5 of Soft Point #15.

Indexing of parameters can also be selected by offsetting the logical number by 128. This is called "vertical indexing." For example, the configuration of:

Start 109 End Addr 114 Type 16 Lgl# 128 Param 2 Conv 1

specifies six parameters for FST number 1 (LGL# 0 + 128) starting at parameter # 2. So register 109 = parameter 2 of FST #1, register 110 = parameter 3 of FST #1, ..., and register 114 = parameter 7 of FST #1.

For another example of vertical indexing, the configuration of:

Start: 57428 End Addr: 57437 Type: 17 Lgl#: 137 Param: 2 Conv: 0

specifies ten parameters for Soft Point number 10 (Lgl# 9 + 128) starting at parameter #2. So register 57428 = parameter 2 of Soft Point #10, register 57429 = parameter 3 of Soft Point #10, ..., and register 57437 = parameter 11 of Soft Point #10.

Another way to explain the Lgl# field, is that when the logical number is 128 or greater, the point number will remain the same, and the parameter within that point will increment. When the logical number is less than 128, the point number will increment, and the parameter number will remain the same.

Logical numbers are zero based. For example, there are 32 Soft Points, with the first being logical 0, and the last being logical 31 for a total of 32. To get the correct logical number for the point you wish to access, subtract 1 from the number. Soft Point 5 is logical 4, and Soft Point 27 is logical 26, Analog Input 3 is logical 2, and so on.

Modbus protocol supports 32 Soft points, so the valid Soft point logical numbers are 0-31, and 128-159. If you have 10 Analog Inputs, the valid logical numbers are 0-9, and 128-137. There are 4 FST's, so the valid logical numbers are 0-3, and 128-131. If you have 7 Pulse Inputs, the valid logical numbers are 0-6, and 128-134.

Param Field -- This field specifies the parameter of the Point Types. Be aware of the different data types (Character, Integer, Long, Float) and the size of the data types. See Appendix A for the complete list of the ROC Point Types and Parameters.

Conv Field -- The convert field specifies the type of conversion required, if any, on the data before it is sent to the host or before it is written to the ROC. The conversions are used to allow integer values to be transmitted and received instead of floating point values. Table 3-1 lists the Convert Codes used with the Modbus Protocol Emulation program.

3.4 MODBUS CONVERT CODE DESCRIPTIONS

The paragraphs following Table 3-1 describe the functions of the Modbus Convert Codes which are applied by the Conv field in the Modbus Function Configuration display.

Table 3-1. Modbus Function Convert Codes

Convert Code	Description	Slave	Host
0	No Conversion	---	---
1	Float to integer, Float Scale 1	3,4	6,16
2	Float to integer, Float Scale 2	3,4	6,16
3	Float to integer, Float Scale 3	3,4	6,16
4	Float to integer, Float Scale 4	3,4	6,16
5	Float to integer, Float Scale 5	3,4	6,16
6	Float to integer, Float Scale 6	3,4	6,16
7	Float to integer, Float Scale 7	3,4	6,16
8	Float to integer, Float Scale 8	3,4	6,16
9 to 16	No Conversion	6,16	3,4
17	Integer to Float, Float Scale 1	6,16	3,4
18	Integer to Float, Float Scale 2	6,16	3,4
19	Integer to Float, Float Scale 3	6,16	3,4
20	Integer to Float, Float Scale 4	6,16	3,4
21	Integer to Float, Float Scale 5	6,16	3,4
22	Integer to Float, Float Scale 6	6,16	3,4
23	Integer to Float, Float Scale 7	6,16	3,4
24	Integer to Float, Float Scale 8	6,16	3,4
25 to 32	No Conversion	---	---
33	Character to Integer	3,4	6,16
34	Integer to Character	6,16	3,4

Table 3-1. Modbus Function Convert Codes (continued)

Convert Code	Description	Slave	Host
35	Long to Integer	3,4	6,16
36	Integer to Long	6,16	3,4
37	Packed Discrete Input Values	3,4	6,16
38	Packed Discrete Output Values	6,16	3,4
39	Float to Integer, No Scaling	3,4	6,16
40	Integer to Float, No Scaling	6,16	3,4
41	Float to Byte, No Scaling	3,4	6,16
42	Byte to Float, No Scaling	6,16	3,4
43	Float to Long, No Scaling	3,4	6,16
44	Long to Float, No Scaling	6,16	3,4
45	Float to Byte	6,16	---
46	Float to Unsigned Integer	6,16	---
47	Float to Unsigned Long	6,16	---
48	No Conversion		
49	Deadband, Float Scale 1	6,16	3,4
50	Deadband, Float Scale 2	6,16	3,4
51	Deadband, Float Scale 3	6,16	3,4
52	Deadband, Float Scale 4	6,16	3,4
53	Deadband, Float Scale 5	6,16	3,4
54	Deadband, Float Scale 6	6,16	3,4
55	Deadband, Float Scale 7	6,16	3,4
56	Deadband, Float Scale 8	6,16	3,4
57 to 64	No Conversion	---	---
65	IEEE Floating Point Number	3,4	16
66	IEEE Floating Point Number	3,4	16

Table 3-1. Modbus Function Convert Codes (continued)

Convert Code	Description	Slave	Host
67	IEEE Floating Point Number	3,4	16
68	IEEE Floating Point Number	3,4	16
69	IEEE Floating Point Number	3,4	16
70	IEEE Floating Point Number	3,4	16
71	IEEE Floating Point Number	3,4	16
72	IEEE Floating Point Number	3,4	16
73	IEEE Floating Point Number	3,4	6, 16
74	IEEE Floating Point Number	3,4	6, 16
75 to 255	No Conversion	---	---

Convert Codes 1 to 8, Float to Integer --The Float to Integer conversion changes ROC floating point data to an integer for transmission to the host. The number of the Convert Code specifies which floating point scaling value is to used for the conversion.

Convert Codes 17 to 24, Integer to Float --The Integer to Float conversion changes a transmitted integer value to a floating point value for the ROC. The number of the Convert Code specifies which floating point scaling value is to be used for the conversion. If no fractional part is coming through on conversion from integer to float, use a float conversion that more closely fits the range of integer to be converted (e.g., float range 0 to 10 instead of 0 to 1000).

Convert Code 33, Character to Integer --The Character to Integer conversion changes a ROC character data type to an integer for transmission to the host.

Convert Code 34, Integer to Character --The Integer to Character conversion changes a transmitted integer value to a character data type for the ROC.

Convert Code 35, Long to Integer --The Long to Integer conversion changes a ROC long data type to an integer for transmission to the host.

Convert Code 36, Integer to Long --The Integer to Long conversion changes a transmitted integer value to a long data type for the ROC.

Convert Code 37, Packed Discrete Input Values --The Packed Discrete Input Values packs the status value of a maximum of 16 discrete inputs. The Start and End Addresses must be the same since only one address is used in the data table. If more discrete inputs are needed, configure another line in

the configuration table with the Lgl# (Logical Number) configured to point to the remaining discrete inputs. If less than 16 discrete inputs are configured in the system, the unused bits in the 16 bit word are set to "0". The first bit (bit 0) corresponds to the specified logical number: bit 1 = Lgl# + 1, bit 2 = Lgl# + 2, and so on.

Convert Code 38, Packed Discrete Output Values --The Packed Discrete Output Values assigns the bits of a transmitted 16 bit word (integer) to the status value of discrete outputs. A maximum of 16 discrete outputs can be configured per line in the configuration table. The Start and End Addresses must be the same since only one address is used in the data table. If more discrete outputs are needed, configure another line in the configuration table with the Lgl# (Logical Number) configured to point to the remaining discrete outputs. If less than 16 discrete inputs are configured in the system, the Convert Code only uses as many bits as needed. The first bit (bit 0) corresponds to the specified logical number: bit 1 = Lgl# + 1, bit 2 = Lgl# + 2, and so on.

Convert Code 39, Float to Integer, No Scaling --The Float to Integer conversion changes a ROC floating point data type to an integer for transmission to the host.

Convert Code 40, Integer to Float, No Scaling --The Integer to Float conversion changes a transmitted integer value to a floating point data type for the ROC.

Convert Code 41, Float to Byte, No Scaling --The Float to Byte conversion changes a ROC floating point data type to a byte for transmission to the host.

Convert Code 42, Byte to Float, No Scaling --The Byte to Float conversion changes a transmitted byte value to floating point data type for the ROC.

Convert Code 43, Float to Long, No Scaling --The Float to Long conversion changes a ROC floating point data type to a Long Integer for transmission to the host.

Convert Code 44, Long to Float, No Scaling --The Long to Float conversion changes a transmitted Long Integer value to a floating point data type for the ROC.

Convert Code 45, Float to Byte, Unsigned Character --The Float to Byte Unsigned Character conversion changes a transmitted floating point value to an unsigned character data type for the ROC.

Convert Code 46, Float to Unsigned Integer --The Float to Unsigned Integer conversion changes a transmitted floating point value to an unsigned integer data type for the ROC.

Convert Code 47, Float to Unsigned Long --The Float to Unsigned Long conversion changes a transmitted floating point value to an unsigned long data type for the ROC.

Convert Code 49 to 56, Deadband --The Deadband code is normally used in the host mode when there is frequent polling of a slave device and logging to the Event Log is selected. The deadband prevents minor changes from being recorded in the Event Log. The Deadband uses the HI Float Scale as the upper limit and the LO Float Scale as the lower limit. If the new value is outside the limits (HI Limit = current value + HI Float Scale, LO Limit = current value - LO Float Scale), the new value is

written to the parameter and is logged in the Event Log (if Log Data = 1). The Deadband function only works on floating point values.

Convert Code 56 to 64, No Conversion

Convert Code 65 to 72, IEEE Floating Point Number -- Convert codes 65 to 72 allow a four byte IEEE formatted floating point number to be sent or received in two Modbus registers with the byte orders configurable and listed below. Since these conversions require two registers, Modbus function 6 is not supported. A check is made to ensure that an even number of registers is requested, that the starting register number does not begin in the middle of a register pair, and that the number of registers does not exceed the number of registers configured.

Byte 0	Byte 1	Byte 2	Byte 3
seee eeee	emmm mmmm	mmmm mmmm	mmmm mmmm

where: s = Sign Bit e = Exponent Bit m = Mantissa Bit

Convert Code 65 -- Places byte 0 and byte 1 in register xxxxx; byte 2 and byte 3 are placed in register xxxxx + 1.

Register xxxxx	byte 0, byte 1
Register xxxxx+1	byte 2, byte 3

Convert Code 66 -- Places byte 0 and byte 1 in register xxxxx; byte 2 and byte 3 are placed in register xxxxx + 1. Same as Convert Code 65 regardless of MSB 1st flag.

Convert Code 67 --Reverses byte 0 and byte 1 order in register xxxxx; reverses byte 2 and byte 3 order in register xxxxx + 1.

Register xxxxx	byte 1, byte 0
Register xxxxx+1	byte 3, byte 2

Convert Code 68 --Reverses byte 0 and byte 1 order in register xxxxx; reverses byte 2 and byte 3 order in register xxxxx + 1. Same as Convert Code 67 regardless of MSB 1st flag.

Convert Code 69 -- Places byte 2 and byte 3 in register xxxxx; byte 0 and byte 1 are placed in register xxxxx + 1.

Register xxxxx	byte 2, byte 3
Register xxxxx+1	byte 0, byte 1

Convert Code 70 -- Places byte 2 and byte 3 in register xxxxx; byte 0 and byte 1 are placed in register xxxxx + 1. Same as Convert Code 69 regardless of MSB 1st flag.

Convert Code 71 --Reverses byte 2 and byte 3 order in register xxxxx; reverses byte 0 and byte 1 order in register xxxxx + 1.

Register xxxxx	byte 3, byte 2
Register xxxxx+1	byte 1, byte 0

Convert Code 72 --Reverses byte 2 and byte 3 order in register xxxxx; reverses byte 0 and byte 1 order in register xxxxx + 1. Same as Convert Code 71 regardless of MSB 1st flag.

Convert Code 73 and 74 -- Convert Codes 73 and 74 send the IEEE formatted floating point number as four bytes with a single register request. Only the byte order is changed. These conversions are supported with Modbus function 6.

Convert Code 73 -- Loads register xxxxx in byte 2, byte 3, byte 0, byte 1 order.

Convert Code 74 -- Loads register xxxxx in byte 2, byte 3, byte 0, byte 1 order regardless of MSB 1st flag.

3.5 MODBUS HOST

The Modbus host mode of operation allows the ROC to simulate a host device that can poll other devices for data and to store the data for parameter updates, for use in FST program registers, and for use as User C variables. The ROC can also send commands to set outputs and to write data to a slave device. The Modbus host mode is enabled by the host Enable field on the Modbus configuration display.

3.5.1 Modbus Host Configuration

The Modbus host configuration parameters are set and edited with the Modbus host Configuration Display. Upon selection of a Modbus host UDP, the GV101 Configuration Software displays the configuration parameters as shown in Figure 3-4. Descriptions of the Modbus host configuration parameters follow the figure.

Up to forty different host requests/commands can be configured for a Comm port. Each request/command can transmit or receive up to 240 bytes of data. Modbus Functions 1, 2, 3, 4, 5, 6, 15, and 16 are supported in the host configuration. Functions 1, 2, 3, and 4 request data from slave devices, and functions 5, 6, 15, and 16 transmit data to a slave device.

Each host request configured must have a corresponding entry in the Modbus Functions configuration. When using Modbus functions 1, 2, 3, and 4, the host will be reading data from the slave and writing it to the host. When using Modbus functions 5, 6, 15, and 16, the host will be reading data from the host and writing it to the slave. Care must be given to insure that there are enough points in the host to support the points being written to or read from in the slaves. Utilization of ROC soft point parameters, unused I/O module parameters, and FST parameters may be required to store all of the necessary data. There is a Modbus Host Configuration worksheet in Appendix E of this manual.

```
                        Modbus Host COM1 1 of 3

                                 Host 0-9

0. RTU Addr 1    Fnc Num 3  Reg# 7000   Save# 7000   #Regs 10   Status 8
1. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
2. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
3. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
4. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
5. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
6. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
7. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
8. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0
9. RTU Addr 0    Fnc Num 0  Reg# 0      Save# 0      #Regs 0    Status 0



1 Update  2 Prev.   3 Next  4 RD Disk  5 WR Disk  6 Quit  7    8 Save
```

Figure 3-4. Modbus Host Configuration Display

RTU Addr -- Specifies the address of the slave device. Address "0" is used to broadcast a command to all devices connected to the port. The address can be a number from 0 to 255.

Fnc Num -- Specifies the Modbus function to be sent to the slave device. The valid function numbers are:

Function 1	Send data to host (logic coils)
Function 2	Send data to host (discrete inputs)
Function 3	Send data to host (holding registers)
Function 4	Send data to host (input registers)
Function 5	Set a single coil in slave device
Function 6	Set a single register in slave device
Function 15	Set multiplecoils in slave device
Function 16	Set multiple registers in slave device

A function number of "0" disables this request/command. Any other function number not described above results in an error (Status = 145) and aborts the request.

Reg# -- A number that represents the starting Data Register (for this request) in the slave device.

Save# -- A number that represents the starting Data Register (for this request) in the host. The Save Register number must be present in the Modbus Functions table for the entered Function Number on the same Comm port. This register tells the host where to put the data when it is received or where to get the data for an output command. This register number does not need to be the same as the device register number (Reg#).

#Regs -- Specifies the number of registers for the host to request or set. The Modbus Functions table for the specified Function Number (Fnc Num) must be sized the same as, or greater than, the #Regs field.

Status -- Displays the status of the host request or command. Table 3-2 lists the possible status values.

Table 3-2. Status of Host Request or Command

Status	Description	Status	Description
0	Inactive or start of transmission	8	Valid slave response
1	Receive timeout error	128	Write ROC Data error
2	Received Address check	129	Access ROC Data error
3	Received Function Number check	130	Host Function Table error
4	Number of expected bytes check	131	Transmit Timeout error
5	Receiving slave response	144	Transmit or Receive buffer overflow
6	CRC or LRC check	145	Invalid Function Number in request
7	CRC or LRC check		

Status values 0 and 2 through 8 are active on the host transmission. These values appear for a very short period of time and then step to the next value if the process is without error. If an error occurs in the step, then the value is present until the next transmission is requested. A transmission without error has a status value of 8.

3.6 MODBUS HOST COMMUNICATION PARAMETERS

The Communications (Comm) Parameters for the port on which Modbus is active determine the way the ROC communicates with other devices. The Modbus host function can be under the control of an FST program. The Comm Port parameters are set with the GV101 "Configure Comm Ports" option. Figure 3-5 shows a typical Comm Port configuration display.

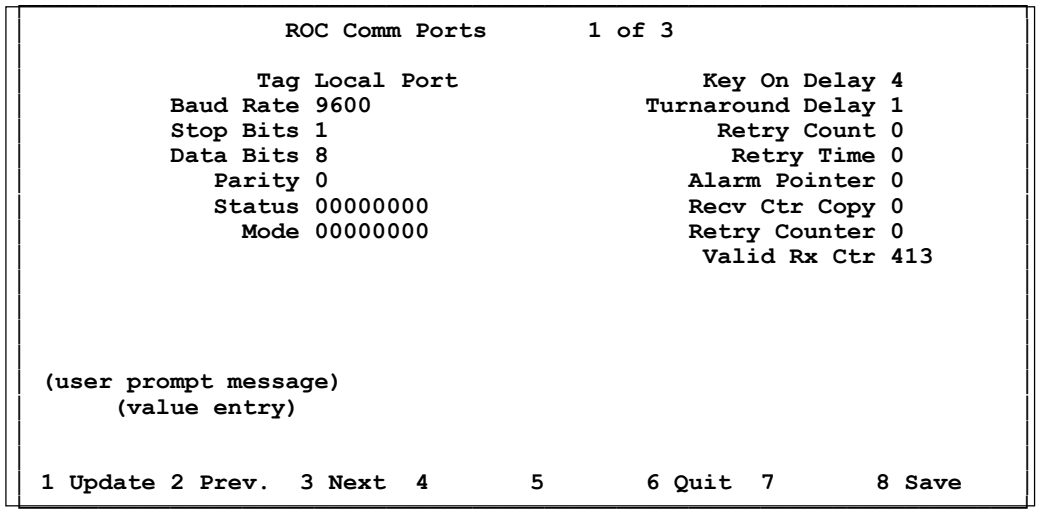

Figure 3-5. Comm Port Configuration Display

The following parameters are used for control of Modbus communications.

Tag -- This parameter identifies the Comm port being configured. An identifier of up to 10 characters will identify the local port, Comm 1, or Comm 2 ports.

Baud Rate -- Transmit and receive data baud rate in bits per second.

Stop Bits -- The number of stop bits contained in a character.

Data Bits -- The number of data bits contained in a character.

Parity -- Parity checks to be performed by the communications controller.

Status -- This is a read-only field, with bits 2 and 7 used for the Modbus host. Bit 7, when on (1), denotes that a Modbus host transmission is in progress. When bit 7 is off (0), the Modbus host program is inactive or waiting for the Retry Time to time out. If bit 2 is on (1), an error has occurred. Bit 2 will remain on (1) until the next successful transmission is finished. Bit 2 will then be set to off (0).

Mode -- Setting of the Mode parameter, Bit 7, to "1" starts a transmission using the request number specified by Retry Count. Subsequent request numbers are transmitted until a Function Number of "0" is encountered or the last request number (39) has been transmitted. At this time, Bit 7 is cleared. Setting Bit 6 of the Mode parameter to "1" along with Bit 7 follows the same sequence as above. Instead of stopping at a "0" function or last request, the host waits the amount of time specified by Retry Time and then starts the sequence over again. This process continues until Bit 7 is set to "0".

Bit 3 is used to enable RTS/CTS handshaking. The Modbus program will assert the RTS signal and wait for the CTS in response. When CTS is detected, the message is sent. (RTS/CTS handshaking is not supported on ROC407 version 1.00.)

Key On Delay -- Push-To-Talk Delay in 50 millisecond intervals for ROC306/312 or ROC364 and 10 millisecond intervals for ROC407 FloBoss. Allows the radio equipment to become fully operational before transmitting. Normal setting would be "4" for the ROC300 Series and "20" for the ROC407 (200 milliseconds).

Turnaround Delay or Key Off Delay -- Allows a delay to be configured for turning off the RTS signal in 50 millisecond intervals for the ROC300 Series and 10 millisecond intervals for ROC407. A Turnaround Delay of "0" will result in the RTS signal being turned off in the middle of the last character transmitted. A Turnaround Delay of "1" or "2" on the ROC 300 Series will turn off the RTS signal approximately 100 milliseconds after the last character is transmitted, and a value of "3" or "4" will turn off RTS after approximately 200 milliseconds, and so on. A Turnaround Delay of "1" on the ROC 407 will turn off the RTS signal 0 to 10 milliseconds after the last character is transmitted, and a value of "2" will turn off RTS after 10 to 20 milliseconds, and so on. The ROC300 Series Modbus user program is limited to a resolution of 100 milliseconds by the ROC Operating System. The ROC407 Modbus user program is scheduled by the operating system as soon as the delay period is completed. On the ROC300 Series only, if a Turnaround Delay greater than or equal to "128" is entered, an extra character will be transmitted, and the RTS signal will be turned off in the middle of the extra character. The extra character will be a LF (hex 0A) if using ASCII mode, and a NULL (hex 00) if using the RTU mode. If the entered value is greater than "128", it will be reset to "128" by the Modbus user program.

When using the following communication cards, a Turnaround Delay of "128" for the ROC300 Series and "2" for the ROC407 should be used rather than "0" to ensure proper control of the RTS signal so that improper termination of the message is avoided:

EIA-422/485 Serial Communications Card
Radio Modem Communications Card
Leased/Private Line Communications Card

Each of the above listed communications cards uses the RTS signal internally to enable its transmit circuitry. If a Turnaround Delay of "0" is used, the RTS signal will be deactivated during the last byte, and the communicating device at the other end will receive an incomplete message. Therefore, a Turnaround Delay value of "128" is recommended so that the extra character is incomplete, but a valid message is still received. When the RTS signal is turned off, the receiver on the communication card is turned on.

When using the following communications cards, a Turnaround Delay of "0" is recommended:

EIA-232 Serial Communications Card
Dial-up Modem Communications Card

To achieve the fastest polling cycles, a Turnaround Delay of "0" or "128" should be used, based on what the connected host or slave at the other end is capable of. Some Modbus hosts or slaves will not tolerate an extra character and some will. The Turnaround Delay should be chosen accordingly.

Retry Count -- This specifies the request number (0 to 39) as configured in the Modbus host Configuration. To begin with, request number "0" indicates the first host request, "1" indicates the second host request, and "39" indicates the fortieth (last) host request. Each request from this number on will be sent until a request is encountered that has a function number of "0".

Retry Time -- This specifies the amount of time, in addition to the normal 500 millisecond interval, to wait between groups of transmissions in 100 millisecond intervals. This parameter is only used if Mode Bit 6 is set to "1".

Alarm Pointer -- A pointer to the last alarm in the alarm log that triggers a report-by-exception (RBX) message (informational only).

Recv Ctr Copy -- The receive counter copy checks the activity of the receive buffer before transmitting an RBX sequence (informational only).

Retry Counter -- The retry counter logs the number of retries in an RBX sequence (informational only).

Valid Rx Ctr -- The valid receive counter logs the number of valid Modbus messages received by the ROC on this communication port. This counter can be preset to a value or cleared.

3.7 MODBUS DIAL-UP OPERATION

The Modbus host can dial up to six different Modbus devices. When attempting to dial, the address specified in the host request is compared to the addresses in modem control to determine the Connect Command to execute. If the address does not match any that are configured, then the first Connect Command is used.

The Modbus host will retry three times before proceeding when an internal or external modem is used for the connection. The Modbus program, before trying to transmit, examines the current state of DSR to determine whether the modem is active or not. If not active, the Modbus program attempts to initialize the modem and reestablish the connection. If an external modem is used, DSR must be configured to indicate whether the modem is active or not. Often, connecting DCD from the modem to DSR for the ROC will work. If the external modem does not supply a signal to DSR indicating whether the modem is active or not, the Modbus program will always determine that the modem is inactive and attempt to re-establish the connection. Refer to Appendix C, Modem Reference, for typical cabling configuration.

3.8 MODEM CONTROL

The Modem Control display (Figure 3-6), accessed through the Configure User Data Types menu, provides the information used to control the operation of a Hayes-compatible modem installed internally or externally.

```
                Modbus Modem COM 1   1 of 1

                           StatusΣ0
              Modem Type (N/H/I) H
                    Connect Time 1000
                 Config Command ATE0V0Q0H0X0&C&S1 S0=1
  RTU Addr 1      Connect Command ATDT2633
  RTU Addr 3      Connect Command ATDT3818
  RTU Addr 0      Connect Command
  RTU Addr 0      Connect Command
  RTU Addr 0      Connect Command
  RTU Addr 2      Connect Command ATDT2418
                  Disconnect Time 80









 1 Update 2      3       4 RD Disk 5 WR Disk 6 QUIT 7        8 SAVE
```

Figure 3-6. Modem Control Display

The Modbus program initializes the modem upon power-up, host modem dial-up, and modem disconnect. Upon initialization, commands are sent to the modem to configure it for proper operation. The Modbus program sends an escape sequence "+++" to place the modem in command mode and then sends the sequence AT[CR] that may be used by the modem to train itself to the communications of the ROC. In addition to these preprogrammed sequences, the user can enter a string to configure the modem for the desired state of operation. In the following example, the Config command string configures the modem for no echo (E0), numeric result codes (V0), display result codes (Q0), go on

hook (H0), basic result codes (X0), data carrier detect always on (&C), data set ready (&S1), and to answer on one ring (S0=1).

ATE0V0Q0H0X0&C&S1 S0=1

Modem configuration is based upon the Hayes AT command set. Appendix C of this manual provides modem reference material. Configuration information for the displayed modem parameters follows.

Status -- Reflects codes returned from the modem.

Modem Type -- Indicates whether a 212 dial-up modem is nonexistent (N), Hayes-compatible located externally (H), or installed internally (I).

Connect Time -- Represents the number of 10-millisecond intervals the program should wait for a connection response from the modem before disconnecting.

Config Command -- Provides a 30-character command line usually used to configure the operation of an internal or external 212 dial-up modem. Some external 212 dial-up modems must be configured by a communication software package that supports the Hayes-compatible command set.

RTU Addr -- Associates an address to the Connect Command. The Modbus program uses the RTU Address to associate host requests to the correct Connect Command when the modem is configured to dial. The RTU Address can have values between 0 and 255.

Connect Command -- Provides a 30-character command line typically used to represent the phone number you wish to dial. The Modbus program uses the RTU Address to associate host requests to the correct Connect Command when the modem is configured to dial.

NOTE

When configuring Config and Connect Commands, consult your modem's user manual for the valid AT command set; otherwise, the modem may not operate as intended.

Disconnect Time -- Represents the number of 10-millisecond intervals following the last valid message received before disconnecting. If the dial is initiated by the ROC, the ROC can stay connected if it receives valid messages from the host. The Disconnect Time begins following the last valid received message while the modem is connected. A modem disconnect time of less than zero will not disconnect the modem.

3.9 CONTROLLING MODBUS HOST BY FST

An FST can be used to schedule Modbus host requests. If an FST is created to schedule host requests, the FST should examine the Status parameter of the communication port and wait until the request is completed before proceeding; otherwise, host requests can be skipped and the desired data transfer may not have completed before proceeding. Modbus host requests set Bit 7 of the Status parameter when a request is attempted. Upon completion of the request, Bit 7 of the Status parameter is cleared. It is good practice to wait for this bit to clear before proceeding.

The FST cannot access errors indicated in the Modbus host Status parameter. The FST needs to employ its own error-checking mechanisms.

Using an FST, other scenarios can be developed to control the Modbus hosts ability to dial other Modbus devices on a regular interval. FST timers can be set to indicate when it is time for the Modbus host to call other Modbus devices. Comparing values for status change, limit exceeded, or deadband exceeded can initiate Modbus host requests to signal warnings to other Modbus devices.

Appendix B provides example FST control schemes.

SECTION 4 — MODBUS MESSAGE FORMAT

This section explains the message formats used in Modbus host/slave communications, the modes of data transmission, and examples of typical transmit and receive messages for the Modbus protocol functions supported by the ROC.

4.1 MODES OF TRANSMISSION

Two modes of transmission are available for use in a Modbus system. Both modes provide the same capabilities for communicating with the Modbus host and are selected depending upon the equipment used as the Modbus host. All devices in the same communication network must be configured with the same mode of transmission.

The two modes of transmission are ASCII and RTU. In the ASCII mode, each character is divided into two 4-bit parts that are represented by their hexadecimal equivalent. In the RTU mode, data is sent in 8-bit binary characters. The ASCII mode uses twice as many characters as the RTU mode.

4.2 MODBUS MESSAGE CONTENTS

The Modbus message consists of the device address, function code, data, error check, and end of frame. In the ASCII transmission mode, the message is sent as ASCII characters representing the hexadecimal characters used to construct the message. The RTU transmission mode sends the message as a continuous 8-bit binary character stream. The following paragraphs define the message fields.

Device Address Field -- The address field immediately follows the beginning of the frame and consists of 8 bits. These bits indicate the user-assigned address of the slave device that is to receive the message sent by the master. The Modbus protocol can support up to 255 remote units on a single communication system. Address “0” is reserved for a broadcast message to all remote units in the ASCII mode. Address “0” is not supported in RTU mode as a slave device.

Data Field -- The data field contains either information needed by the slave to perform the specified function, or it contains data collected by the slave in response to a query. All register/coil address references in Modbus messages are numbered relative to “0” and are actually an offset from the beginning of the register/coil area that can be referenced by the function specified in the message.

Error Check Field -- This field detects errors in transmission. For the RTU mode, the error check field uses a Cyclic Redundancy Check (CRC-16) and requires two 16-bit bytes. For the ASCII mode, the error check field uses the Longitudinal Redundancy Check (LRC) method and requires two ASCII bytes.

Function Code Field -- The Function Code is a single-byte field that defines the action to be performed by the slave. The high-order bit of this field is set by the slave if an error is detected in the received
message. This bit should remain zero if the message is a normal response message. The ROC supports the Modbus function codes listed in Table 1-1.

4.2.1 ASCII Message Framing

Framing, in the ASCII transmission mode, uses the colon (:) character to indicate the beginning of the frame and a carriage return (CR) followed by a line feed (LF) to delineate the end of the frame. The line feed character also serves as a synchronizing character indicating that the transmitting station is ready to receive an immediate reply. Table 4-1 shows the ASCII message format.

Table 4-1. ASCII Message Format

Beginning of Frame	Address	Function	Data	Error Check	End of Frame	Ready to Respond
:	2 Characters 16 Bits	2 Characters 16 Bits	N x 4 Characters 16 Bits	2 Characters 16 Bits	Carriage Return	Line Feed

The error check used in ASCII mode is the LRC method. The error check is an 8-bit binary number represented by, and transmitted as, two ASCII characters. Refer to the *Modbus Protocol Reference Guide (January 1985)* for more detail.

4.2.2 Remote Terminal Unit (RTU) Message Framing

Framing, in the RTU transmission mode, uses the ROC address to indicate the beginning of the frame, and the Error Check to indicate the end of the frame. The Modbus User program uses the received Modbus function number and register number to determine how many bytes of data to receive before the two bytes of CRC. The number of data bytes to receive is determined while receiving the message by looking in the configuration table for the Modbus function, locating a register number match, and gathering the data type. Table 4-2 shows the RTU message format.

Table 4-2. RTU Message Format

Address	Function	Data	Error Check
8 Bits	8 Bits	N x 8 Bits	16 Bits

The Modbus RTU protocol applies a cyclical redundancy check on the message string to produce a 16-bit remainder. This remainder is referred to as the CRC-16 code. The CRC-16 code is then appended to the end of the message string. On receipt of the message, the error check sequence is again performed and if the result is the same, the message contained no transmission errors. Refer to the *Modbus Protocol Reference Guide (January 1985)* for more detail on the CRC-16 code generation.

4.3 EXPLANATION OF FUNCTIONS

The Modbus functions implemented in the ROC are described in the following paragraphs along with examples of their usage.

4.3.1 Function Codes 01 and 02 - Read Output and Input Status

The host uses these function codes to obtain the discrete input values from the ROC. A maximum of 64 discrete inputs can be installed in the ROC and can be obtained in one request. Function Code 02 values are normally read-only input status values, while Function Code 01 values can be written to by another function code.

The host in the following example (Table 4-3) requests eighteen status inputs, starting at address (offset) 1001, from Unit 1. Table 4-4 shows the ROC response message to the host request.

Table 4-3. Function Code 01 Host Request Example Message

Message Field	Device Address	Function Code	Coil/DI Offset	Number of Reads	Error Check
Bytes	1	1	2	2	2
TX Order			MS LS	MS LS	LS MS
Value	01H	01H	03H E9H	00H 12H	CRC-16

Table 4-4. Function Code 01 ROC Response Example Message

Message Field	Device Address	Function Code	Byte Count	Data	Error Check
Bytes	1	1	1	(Byte Count)	2
TX Order					LS MS
Value	As Received	01H	03H		CRC-16

NOTE

The Byte Count defines the number of bytes of discrete data to follow, and is equal to one eighth the number of discrete inputs requested.

The status is packed with one bit for each discrete input. The-low order bit of the first status byte contains the status of the first discrete input; the remaining discrete input statuses follow, going from low-order bit to high-order bit and from the first status byte to the Nth status byte. For discrete input quantities that are not even multiples of eight, the remaining bits in the last status byte are filled with zeroes.

4.3.2 Function Codes 03 and 04 - Read Output and Input Registers

The host uses these function codes to obtain the analog, discrete, PID, AGA, history, and register values from the ROC. A maximum of 255 bytes of inputs can be obtained per request. Function Code 04 is normally used for multiple requests and modified requests. Function Code 03 is normally used for stacked register configuration to allow a single request for multiple data types.

The host in the following example (Table 4-5) requests five values, starting at address (offset) 7001, from Unit 1. The ROC responds (Table 4-6) by sending five floating point values. The Byte Count is twenty, four bytes per value.

Table 4-5. Function Code 03 Host Request Example Message

Message Field	Device Address	Function Code	Register Offset	Number of Reads	Error Check
Bytes	1	1	2	2	2
TX Order			MS LS	MS LS	LS MS
Value	01H	03H	1BH 59H	00H 05H	CRC-16

Table 4-6. Function Code 03 ROC Response Example Message

Message Field	Device Address	Function Code	Byte Count	Data	Error Check
Bytes	1	1	1	(Byte Count)	2
TX Order				(Selectable)	LS MS
Value	As Received	03H	14H		CRC-16

NOTE

The Byte Count defines the number of bytes of register data to follow, and depends on the type of data requested.

4.3.3 Function Code 05 - Force a Single Coil

This packet structure allows the host to force a single coil (Function Code 05) to either ON or OFF. Broadcast mode is allowed by Function Code 5.

The host in the following example (Table 4-7) requests Unit 1 to set a coil at address (offset) 1001. The ROC response (Table 4-8) to the command request is to re-transmit the message as received after the state of the coil has been verified. In Broadcast Mode there would be no response.

Table 4-7. Function Code 05 Host Request Example Message

Message Field	Device Address	Function Code	Value Offset	Data	Error Check
Bytes	1	1	2	2	2
TX Order			MS LS	MS LS	LS MS
Value	01H	05H	03H E9H	FFH 00H	CRC-16

Table 4-8. Function Code 05 ROC Response Example Message

Message Field	Device Address	Function Code	Value Offset	Data	Error Check
Bytes	1	1	2	2	2
TX Order			MS LS	MS LS	LS MS
Value	01H	05H	03H E9H	FFH 00H	CRC-16

The data sent by Function Code 5 to set or clear a coil is:

	MSB	LSB
Clear	00H	00H
Set	FFH	00H

4.3.4 Function Code 06 - Preset a Single Register

Function Code 06 allows the Host to preset a single register. The register acted upon is defined by the address and parameter configured by the MODCFG.EXE program.

The host in the following example (Table 4-9) requests Unit 1 to set a register, at address (offset) 7001, to a floating point value of 100. Broadcast Mode is allowed for Function Code 6. Set the device address to 00 to initiate Broadcast Mode. All devices attached to the network will have the register set to the new value.

The ROC response (Table 4-10) to the command request is to re-transmit the message as received after the register has been modified. In the Broadcast Mode, there will be no response.

Table 4-9. Function Code 06 Host Request Example Message

Message Field	Device Address	Function Code	Value Offset	Data	Error Check
Bytes	1	1	2	4	2
TX Order			MS LS	LS 2 3 MS	LS MS
Value	01H	06H	1BH 59H	00 00 C8 42	CRC-16

Table 4-10. Function Code 06 ROC Response Example Message

Message Field	Device Address	Function Code	Value Offset	Data	Error Check
Bytes	1	1	2	4	2
TX Order			MS LS	LS 2 3 MS	LS MS
Value	01H	06H	1BH 59H	00 00 C8 42	CRC-16

4.3.5 Function Code 15 - Force Multiple Coils

This function code allows the Host to change the state of a block of consecutive coils. The Host can transfer data more efficiently with Function Code 15 than with Function Code 04. There can be up to 64 coil outputs with Function Code 15.

The host in the following example (Table 4-11) requests Unit 1 to set two coils, starting at address (offset) 1001, to the "ON" state. Broadcast Mode is allowed for Function Code 15. Set the device address to 00 to initiate Broadcast Mode. All devices attached to the network will have the selected coils set to "ON".

The ROC response (Table 4-12) to the command request is an echo of the device address, function code, starting address, and quantity of coils set. In the Broadcast Mode, there will be no response.

Table 4-11. Function Code 15 Host Request Example Message

Message Field	Device Address	Function Code	Value Offset	Quantity	Byte Count	Data	Error Check
Bytes	1	1	2	2	1	(Byte Cnt)	2
TX Order			MS LS	MS LS			LS MS
Value	01H	0FH	03H E9H	00H 02H	01H	03H	CRC-16

Table 4-12. Function Code 15 ROC Response Example Message

Message Field	Device Address	Function Code	Value Offset	Quantity	Error Check
Bytes	1	1	2	2	2
TX Order			MS LS	MS LS	LS MS
Value	01H	0FH	03H E9H	00H 02H	CRC-16

NOTE

The Broadcast Mode sets all specified coils active. Unused coils specified will be "ON" and their outputs will be active. Use caution when attempting to service devices with unused coils.

NOTE

The byte count indicates the number of bytes of register or coil data to follow, and is equal to one eighth the number of coils requested for Function Code 15.

For Function Code 15, the status is packed with one bit for each coil. The low-order bit of the first status byte contains the status of the first coil; the remaining coil statuses follow, going from low-order bit to high-order bit and from the first status byte to the Nth status byte. For coil quantities that are not even multiples of eight, the remaining bits in the last status byte are filled with zeroes.

4.3.6 Function Code 16 - Preset Multiple Registers

This function code allows the host to change values in registers. The host can transfer data more efficiently with Function Code 16 than with Function Code 06. There can be up to 60 registers of outputs with Function Code 16. Broadcast Mode is allowed by Function Code 16.

The host in the following example (Table 4-13) requests Unit 1 to set a register, starting at address (offset) 7001, to a floating point value of 100. Broadcast Mode is allowed for Function Code16. Set the device address to 00 to initiate Broadcast Mode. All devices attached to the network will have the register set to the new value.

The ROC response (Table 4-14) to the command request is an echo of the device address, function code, starting address, and quantity of coils set. In the Broadcast Mode, there will be no response.

Table 4-13. Function Code 16 Host Request Example Message

Message Field	Device Address	Function Code	Value Offset	Quantity	Byte Count	Data	Error Check
Bytes	1	1	2	2	1	(Byte Cnt)	2
TX Order			MS LS	MS LS			LS MS
Value	01H	10H	1BH 59H	00H 01H	04H	00 00 C8 42	CRC-16

Table 4-14. Function Code 16 ROC Response Example Message

Message Field	Device Address	Function Code	Value Offset	Quantity	Error Check
Bytes	1	1	2	2	2
TX Order			MS LS	MS LS	LS MS
Value	01H	10H	1BH 59H	00H 01H	CRC-16

4.3.7 Exception Response

The Exception Response message (Table 4-15) is transmitted if the ROC detects an error in the received message. The received function code is returned with the most-significant bit set to signify an error. The exception codes listed in Table 4-16 are for reference only. Although these are the codes specified in the Modbus protocol, some vendors use a subset of these codes or a completely different set of codes.

Table 4-15. ROC Exception Response Example Message

Message Field	Device Address	Function Code	Exception Code	Error Check
Bytes	1	1	1	2
TX Order				LS MS
Value	As Received	FC + 80H	1 to 3	CRC-16

Table 4-16. Modbus Protocol Errors that Elicit an Exception Response Message

Code	Error Detected	Usage
01	Illegal function code	Implemented
02	Illegal Data Address	Implemented
03	Illegal Data Value	Implemented

4.4 SPECIAL FUNCTIONS

Special functions or extensions to Modbus are implemented using Function Code 03, Function Code 04, and Function Code 16. Examples of retrieving historical data, an address table, and discrete output functions follow.

4.4.1 Historical Data Storage

Historical data is contained in Point Type 42 for ROC300-Series units with a ROCPAC, or in Point Type 62 for the ROC407 and ROC300-Series units with a FlashPAC. The ROC can store up to 35 days of hourly data for each of its history points. The historical information is stored in 104-byte blocks of hourly values, one for each daily record. A byte breakdown of the daily record is shown in Table 4-17.

Table 4-17. History Data

Data Byte	Description	Data Byte	Description
0	Valid check, hour 0-7	48 to 51	Hour 11 History Value
1	Valid check, hour 8-15	52 to 55	Hour 12 History Value
2	Valid check, hour 16-23	56 to 59	Hour 13 History Value
3	Not Used	60 to 63	Hour 14 History Value
4 to 7	Hour 0 History Value (12:00 to 1:00 a.m.)	64 to 67	Hour 15 History Value
8 to 11	Hour 1 History Value	68 to 71	Hour 16 History Value
12 to 15	Hour 2 History Value	72 to 75	Hour 17 History Value
16 to 19	Hour 3 History Value	76 to 79	Hour 18 History Value
20 to 23	Hour 4 History Value	80 to 83	Hour 19 History Value
24 to 27	Hour 5 History Value	84 to 87	Hour 20 History Value
28 to 31	Hour 6 History Value	88 to 91	Hour 21 History Value
32 to 35	Hour 7 History Value	92 to 95	Hour 22 History Value
36 to 39	Hour 8 History Value	96 to 99	Hour 23 History Value
40 to 43	Hour 9 History Value	100 to 103	Daily Total or Daily
44 to 47	Hour 10 History Value		Average History Value

Data bytes 0, 1, and 2 are data Valid check bytes. The Valid check bytes provide a flag that denotes if the data for a certain hour is valid. If a bit is set "1", the data is valid. If a bit is clear ("0"), either the data is corrupt or not in the database. The bit designations are:

Data Byte 0

Bit	7	6	5	4	3	2	1	0
Hour	7	6	5	4	3	2	1	0

Data Byte 1

Bit	7	6	5	4	3	2	1	0
Hour	15	14	13	12	11	10	9	8

Data Byte 2

Bit	7	6	5	4	3	2	1	0
Hour	23	22	21	20	19	18	17	16

Due to limitations imposed by available memory in a ROC300-Series controller, only 11 history points can be configured for 35 days of hourly values in base RAM. The user is allowed to configure up to 30 history points in base RAM, but a reduction in number of days of stored values takes place when more than 11 points are configured. RAM areas 1 and 2 each allow a maximum of 30 points with 35 days of hourly values to be stored. Table 4-18 shows the history point numbers for each area of database RAM.

Table 4-18. History Point Numbers for Database RAM

RAM Area	History Point
Base RAM	0 to 29
RAM1	30 to 59
RAM2	60 to 89

The ROC407 allows the use of 30 database points in the base RAM area and 20 database points in RAM area 1, for a total of 50 database points. The ROC407 places no restrictions on the number of days of hourly values in base RAM and allows 35 days of hourly values or 840 entries for all 50 database points.

Historical data can be retrieved by using Function Code 04 and by specifying the month and the day of the history database. No parameter number is required. The Value offset determines which point is accessed in the history database. In the following example for a ROC300 series (Table 4-19), the host requests data for July 23rd from Unit 1 at Database Point 7001. The point to be accessed is the requested address minus the Start Address. If the Start Address for Function Code 04, Point Type 42 is configured as 7001, Database Point "0" would be accessed.

Table 4-19. Host Request for History Data Example Message

Message Field	Device Address	Function Code	Value Offset	Month	Day	Error Check
Bytes	1	1	2	1	1	2
TX Order			MS LS			LS MS
Value	01H	04H	1BH 59H	07H	17H	CRC-16

The ROC will respond with a message as shown in Table 4-20. The 104 byte data message contains the information for the day and month requested. The ROC will return all zeros if no data is available for the day and month requested.

Table 4-20. ROC Response for History Data Example Message

Message Field	Device Address	Function Code	Byte Count	Data	Error Check
Bytes	1	1	1	(Byte Count)	2
TX Order				(Selectable)	LS MS
Value	As Received	04H	68H		CRC-16

4.4.2 Address Table

Point Type 44 for ROC300-Series units with a ROCPAC, or Point Type 64 for the ROC407 and ROC300-Series units with a FlashPAC, allows the transmission of the input data start register, output data start register, high integer scale, and low integer scale values. A total of 4 registers must be specified.

4.4.3 Discrete Outputs

Discrete outputs have a special case when Function Code 16 is used with Parameter 1 (Time On) selected and a floating point value is written to Parameter 1. The floating point value has different effects on the discrete output. They are:

Value	Effect
= 0	Discrete output latched off.
= 1	Discrete output latched on.
> 1	Discrete output momentary on, value is in seconds.
< 0	No effect.
> 0 and < 1	No effect.

In order to utilize this capability, the Modbus function table must be configured with the following parameters:

```
                          Modbus Funct COM1

                          TAG ID FUNCT 16B

1  Start 100   End Addr 100   Type 2    Lgl# 0      Param 1     Conv 46
     •             •            •         •           •           •
     •             •            •         •           •           •
     •             •            •         •           •           •

1 Update   2 Prev.   3 Next   4 RD Disk   5 WR Disk   6 Quit   7     8 Save
```

This page is intentionally blank.

SECTION 5 — PROBLEM SOLVING

5.1 ROC FAILS TO RESPOND TO HOST REQUESTS

If the ROC does not respond to a Modbus request, check to see that:

1) Power is applied.
2) The ROC is plugged into the correct host communications port.
3) The ROC is cabled correctly.
4) The station address is correct.
5) The baud rate is properly set.
6) The number of data bits and stop bits is correctly set.
7) The ROC communications port is configured correctly.
8) The proper communication cards are installed and in the correct order (refer to the appropriate ROC instruction manual).

5.2 ROC ERROR MESSAGES

Error messages are usually caused by the ROC configuration. Check to see that:

1) The proper format (ASCII or RTU) is selected.
2) The Modbus Functions are configured correctly.
3) There are enough continuous registers configured.
4) The requested data types match the configured data types.
5) The conversions are used correctly.

5.3 ROC RESPONDS WITH WRONG DATA

Wrong data is usually caused by the ROC configuration. Check to see if:

1) The data order is wrong. Check MSB and LSB configuration.
2) The wrong registers are configured.
3) The wrong parameter is selected.
4) Indexing is not proper. See Section 3.3, Lgl# field.

5.4 ROC APPENDS A CHARACTER TO MESSAGE

Check to see if a turnaround delay is configured in the Communication Parameters when none is needed.

5.5 ROC DOES NOT RESPOND IN POINT TO MULTI-POINT COMM SYSTEM

Check to see if the ROC responds in a point-to-point hook-up. If it does, then the problem is associated with delays in the system. Check see that:

1) The ROC key-on delay has been configured correctly.
2) The ROC turnaround delay has been configured correctly.
3) The host has configurable key-on and turnaround delays.
4) Each ROC has a unique address.

5.6 DELETING USER-DEFINED POINTS

User-Defined Points can be deleted by first disabling the user program that defines the User-Defined Point and then performing a cold hard start. The cold hard start initializes the database and removes the User-Defined Point definitions, providing they have not previously been saved to EEPROM.

5.7 INTEGER-TO-FLOAT CONVERSION

Integer-to-float conversions may not produce a fractional component. To correct, use a float scale range that more closely corresponds to the range of the integer being converted.

5.8 ROC RESPONSE PRECEDED BY MODEM COMMANDS

If the ROC sends modem commands before the transmission of a response to a request, connect the ROC's DSR signal to the modem's DCD signal or a similar pin indicating modem activity. The DSR of some modems can be configured to indicate modem activity.

5.9 DELETING A USER PROGRAM FROM ROC407 FLASH MEMORY

User programs loaded in the flash memory of a ROC407 FloBoss are not deleted in the same way user programs loaded in RAM would be deleted. Programs loaded in RAM are removed with the Clear All User Memory selection under User Program Routines in the Utilities menu of GV101. This routine has no effect on user programs loaded in flash memory.

To clear programs loaded into flash memory, you must download a "dummy" user program containing all zeros to the affected blocks of flash memory. Two such programs, FBCLR_A0.H00 and FBCLR_A8.H00, are supplied with the other Modbus user program files. When downloaded, FBCLR_A0.H00 will clear out any program in the A000 and A400 blocks; FBCLR_A8.H00 will clear out any program in the A800 and AC00 blocks. To download these programs using the GV101 configuration software, select Utilities, then User Program Routines, and then Program Flash Memory. Refer to the procedure in Section 2.4 for information on how to select a file and download it. At the end of the procedure, instead of enabling the associated user flag, you should ensure that it is **disabled** (set to "0"). A Warm Start is not required.

This page intentionally blank.

APPENDIX A — ROC POINT TYPES

A.1 ROC POINT AND DATA TYPES

Point types and data types for the Remote Operations Controller (ROC) are shown in Tables A-1 and A-2.

Table A-1. Valid ROC Point Types

Type	Description	Type	Description
0	Configurable Opcode	15	System Variables
1	Discrete Inputs	16	FST Registers
2	Discrete Outputs	17	Soft Points
3	Analog Inputs	18	Reserved
4	Analog Outputs	19	Database Setup
5	Pulse Inputs	20	ROC Tasks
6	PID Control	21	Reserved
7	AGA Flow Parameters	22, 23	User Defined
8	Reserved	24	Reserved
9	Local Display Panel (ROC300 w/ROCPAC only)	25 to 39	User Defined
10	AGA Flow Values	40	MVS (205 Sensor) Parameters (ROC407 only)
11	Tank Parameters (ROC300 w/ROCPAC only)	41	AGA Run Parameters (not for ROC300 w/ROCPAC)
12	ROC Clock	42	Extra AGA Run Parameters (not for ROC300 w/ROCPAC)
13	System Flags	43	User Lists (ROC407 only)
14	Communications Ports	44 to 60	Reserved (ROC407 only)

The point numbers used for the physical I/O (point types 1 to 5) are:

Point numbers 0 to 63 are assigned as the field I/O (modular or built-in I/O).
Point numbers 64 to 68 are assigned as the diagnostic (system) I/O.

Table A-2. Data Types

Data Type	Definition
BN	binary
AC	ASCII character
SC	signed character
UC	unsigned character
SI	signed integer
UI	unsigned integer
SL	signed long integer
UL	unsigned long integer
FL	float point (IEEE Format)
TLP	Type, Point or Logical Number, Parameter Number

A.2 ROC POINT PARAMETER DEFINITIONS

Tables A-3 through A-25 show the parameter definitions for each point type listed in Table A-1. For further descriptions of the parameters used in configuring the ROC, refer to the *Type GV101 Configuration Software User Manual (Form A4194).*

Table A-3. Point Type 0, Configurable Opcode

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	FL	4	Sequence/rev #
1	R/W	TLP	3	Data 1
2	R/W	TLP	3	Data 2
3	R/W	TLP	3	Data 3
4	R/O	TLP	3	Data 4
5	R/W	TLP	3	Data 5
6	R/W	TLP	3	Data 6
7	R/W	TLP	3	Data 7
8	R/W	TLP	3	Data 8
9	R/W	TLP	3	Data 9
10	R/W	TLP	3	Data 10
11	R/W	TLP	3	Data 11
12	R/W	TLP	3	Data 12
13	R/W	TLP	3	Data 13
14	R/W	TLP	3	Data 14
15	R/W	TLP	3	Data 15
16	R/W	TLP	3	Data 16
17	R/W	TLP	3	Data 17
18	R/W	TLP	3	Data 18
19	R/W	TLP	3	Data 19
20	R/W	TLP	3	Data 20
21	R/W	TLP	3	Data 21
22	R/O	TLP	3	Data 22
23	R/W	TLP	3	Data 23
24	R/W	TLP	3	Data 24
25	R/W	TLP	3	Data 25
26	R/W	TLP	3	Data 26
27	R/O	TLP	3	Data 27
28	R/W	TLP	3	Data 28
29	R/W	TLP	3	Data 29
30	R/W	TLP	3	Data 30
31	R/W	TLP	3	Data 31
32	R/W	TLP	3	Data 32
33	R/W	TLP	3	Data 33
34	R/W	TLP	3	Data 34
35	R/W	TLP	3	Data 35
36	R/W	TLP	3	Data 36
37	R/W	TLP	3	Data 37
38	R/W	TLP	3	Data 38

Table A-3. Point Type 0, Configurable Opcode (continued)

Parm Number	Read-Write	Data Type	Length	Description
39	R/W	TLP	3	Data 39
40	R/W	TLP	3	Data 40
41	R/W	TLP	3	Data 41
42	R/W	TLP	3	Data 42
43	R/W	TLP	3	Data 43
44	R/W	TLP	3	Data 44

Table A-4. Point Type 1, Discrete Input Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id
1	R/W	UC	1	Filter
2	R/W	UC	1	Status
3	R/W	UC	1	Mode
4	R/O	UC	1	Alarm Code
5	R/W	UL	4	Accumulated Value
6	R/W	UL	4	On Counter
7	R/W	UL	4	Off Counter
8	R/W	UI	2	0% Count
9	R/W	UI	2	100% Count
10	R/W	UI	2	Max Sample Time
11	R/W	AC	10	Units
12	R/W	UI	2	Scan Period
13	R/W	FL	4	Low Reading EU
14	R/W	FL	4	High Reading EU
15	R/W	FL	4	Low Alarm EU
16	R/W	FL	4	High Alarm EU
17	R/W	FL	4	Lo Lo Alarm EU
18	R/W	FL	4	Hi Hi Alarm EU
19	R/W	FL	4	Rate Alarm EU
20	R/W	FL	4	Alarm Deadband
21	R/W	FL	4	EU Value
22	R/O	UI	2	TDI Count

Table A-5. Point Type 2, Discrete Output Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id.
1	R/W	UI	2	Time On
2	R/O	UC	1	(Not used)
3	R/W	UC	1	Status
4	R/W	UC	1	Mode
5	R/O	UC	1	Alarm Code
6	R/W	UL	4	Accumulated Value
7	R/W	AC	10	Units
8	R/W	UI	2	Cycle Time
9	R/W	UI	2	0% Count
10	R/W	UI	2	100% Count
11	R/W	FL	4	Low Reading EU
12	R/W	FL	4	High Reading EU
13	R/W	FL	4	EU Value

Table A-6. Point Type 3, Analog Input Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id.
1	R/W	AC	10	Units
2	R/W	UI	2	Scan Period
3	R/W	UI	2	Filter
4	R/W	UI	2	Adjusted A/D 0%
5	R/W	UI	2	Adjusted A/D 100%
6	R/W	FL	4	Low Reading EU
7	R/W	FL	4	High Reading EU
8	R/W	FL	4	Low Alarm EU
9	R/W	FL	4	High Alarm EU
10	R/W	FL	4	Lo Lo Alarm EU
11	R/W	FL	4	Hi Hi Alarm EU
12	R/W	FL	4	Rate Alarm EU
13	R/W	FL	4	Alarm Deadband
14	R/W	FL	4	Filtered EUs
15	R/W	UC	1	Mode
16	R/O	UC	1	Alarm Code
17	R/O	UI	2	Raw A/D Input
18	R/O	UI	2	Actual Scan Time

Table A-7. Point Type 4, Analog Output Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id.
1	R/W	AC	10	Units
2	R/W	UI	2	Adjusted D/A 0%
3	R/W	UI	2	Adjusted D/A 100%
4	R/W	FL	4	Low Reading EU
5	R/W	FL	4	High Reading EU
6	R/W	FL	4	Value in EUs
7	R/W	UC	1	Mode
8	R/O	UC	1	Alarm Code
9	R/O	UI	2	Raw D/A Output

Table A-8. Point Type 5, Pulse Input Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id.
1	R/W	AC	10	Units
2	R/W	UC	1	Rate Flag
3	R/W	UC	1	Rate Period
4	R/O	UC	1	Type
5	R/W	UI	2	Scan Period
6	R/W	FL	4	Conversion
7	R/W	FL	4	Lo Alarm EU
8	R/W	FL	4	Hi Alarm EU
9	R/W	FL	4	Lo Lo Alarm EU
10	R/W	FL	4	Hi Hi Alarm EU
11	R/W	FL	4	Rate Alarm EU
12	R/W	FL	4	Deadband or Rollover
13	R/W	FL	4	Value in EUs
14	R/W	UC	1	Mode
15	R/O	UC	1	Alarm Code
16	R/W	UL	4	Accumulated Value
17	R/O	FL	4	Current Rate
18	R/W	FL	4	Today's Total
19	R/O	FL	4	Yesterday's Total

Table A-9. Point Type 6, PID Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id.
1	R/W	UC	1	Control Type
2	R/O	UC	1	Loop Status
3	R/O	UI	2	Scan Time
4	R/W	TLP	3	Primary Input Definition
5	R/W	TLP	3	Primary Output Definition
6	R/W	FL	4	Primary Switch Setpoint
7	R/W	TLP	3	Primary Switch Process Variable Definition
8	R/W	AC	1	Primary Switch Mode
9	R/W	TLP	3	Override Input Definition
10	R/W	TLP	3	Override Output Definition
11	R/W	FL	4	Override Switch Setpoint
12	R/W	TLP	3	Override Switch Process Variable Definition
13	R/W	AC	1	Override Switch Mode
14	R/W	FL	4	Primary Setpoint
15	R/W	FL	4	Primary Setpoint EU/min
16	R/W	UI	2	Primary Loop Period
17	R/W	FL	4	Primary Proportional Gain
18	R/W	FL	4	Primary Integral Gain
19	R/W	FL	4	Primary Derivative Gain
20	R/W	FL	4	Primary Scale Factor
21	R/W	FL	4	Primary Integral Deadband
22	R/W	FL	4	Primary Process Variable
23	R/W	FL	4	Primary Output EU
24	R/W	FL	4	Primary Switch Process Variable
25	R/W	UI	2	Minimum Control Time
26	R/W	FL	4	Override Setpoint
27	R/W	FL	4	Override Setpoint EU/min
28	R/W	UI	2	Override Loop Period
29	R/W	FL	4	Override Proportional Gain
30	R/W	FL	4	Override Integral Gain
31	R/W	FL	4	Override Derivative Gain
32	R/W	FL	4	Override Scale Factor
33	R/W	FL	4	Override Integral Deadband
34	R/W	FL	4	Override Process Variable
35	R/W	FL	4	Override Output EU
36	R/W	FL	4	Override Switch Process Variable

Table A-10. Point Type 7, AGA Flow Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Tag Id.
1	R/W	FL	4	Latitude
2	R/W	FL	4	Elevation
3	R/W	BN	1	Calculation Method
4	R/W	BN	1	AGA Configuration
5	R/W	FL	4	Specific Gravity
6	R/W	FL	4	Heating Value
7	R/W	FL	4	Gravitational Accel.
8	R/W	UI	2	Scan Period
9	R/W	FL	4	Pipe Diameter
10	R/W	FL	4	Orifice Diameter
11	R/W	FL	4	Orifice Reference Temp.
12	R/W	UC	1	Orifice Material
13	R/W	AC	30	Meter Run Id.
14	R/O	UC	1	Alarm Code
15	R/W	FL	4	Low Alarm
16	R/W	FL	4	High Alarm
17	R/W	FL	4	Viscosity
18	R/W	FL	4	Specific Heat Ratio
19	R/W	FL	4	Base Pressure
20	R/W	FL	4	Base Temperature
21	R/W	FL	4	Low Flow Cutoff
22	R/W	FL	4	Fpwl (Gravitational correction factor)
23	R/W	FL	4	N2 Nitrogen
24	R/W	FL	4	CO2 Carbon Dioxide
25	R/W	FL	4	H2S Hydrogen Sulfide
26	R/W	FL	4	H2O Water
27	R/W	FL	4	He Helium
28	R/W	FL	4	CH4 Methane
29	R/W	FL	4	C2H6 Ethane
30	R/W	FL	4	C3H8 Propane
31	R/W	FL	4	C4H10 n-Butane
32	R/W	FL	4	C4H10 i-Butane
33	R/W	FL	4	C5H12 n-Pentane
34	R/W	FL	4	C5H12 i-Pentane
35	R/W	FL	4	C6H14 n-Hexane
36	R/W	FL	4	C7H16 n-Heptane
37	R/W	FL	4	C8H18 n-Octane
38	R/W	FL	4	C9H20 n-Nonane
39	R/W	FL	4	C10H22 n-Decane

Table A-10. Point Type 7, AGA Flow Parameters (continued)

Parm Number	Read-Write	Data Type	Length	Description
40	R/W	FL	4	O2 Oxygen
41	R/W	FL	4	CO Carbon Monoxide
42	R/W	FL	4	H2 Hydrogen
43	R/O	UC	1	(Not used)
44	R/W	UC	1	Stacked Dp Enable
45	R/W	TLP	3	Low Dp Input
46	R/W	TLP	3	Differential Pressure Input
47	R/W	TLP	3	Static Pressure Input
48	R/W	TLP	3	Temperature Input
49	R/W	FL	4	Low Dp Setpoint
50	R/W	FL	4	High Dp Setpoint
51	R/W	FL	4	Meter Value
52	R/W	FL	4	Static Pressure Value
53	R/W	FL	4	Temperature Value

Table A-11. Point Type 9, Local Display Panel (LDP) Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Text for Line 1
1	R/W	AC	10	Text for Line 2
2	R/W	AC	10	Text for Line 3
3	R/W	TLP	3	Data for Line 1
4	R/W	TLP	3	Data for Line 2
5	R/W	TLP	3	Data for Line 3

Table A-12. Point Type 10, AGA Flow Value (Calculation) Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/O	FL	4	Meter Value
1	R/O	FL	4	Static Pressure Value
2	R/O	FL	4	Temperature Value
3	R/O	FL	4	Instantaneous Flow
4	R/O	FL	4	Instantaneous Energy
5	R/W	FL	4	Flow Today
6	R/W	FL	4	Energy Today
7	R/W	FL	4	Flow Yesterday
8	R/W	FL	4	Energy Yesterday
9	R/O	FL	4	Pressure Extension (hpwf)
10	R/O	FL	4	C Prime (C')
11	R/O	FL	4	Sample Time
12	R/O	FL	4	Expansion Factor
13	R/O	FL	4	Fr
14	R/O	FL	4	Ftf
15	R/O	FL	4	Fpv
16	R/O	FL	4	Fgr
17	R/O	FL	4	Fb
18	R/O	FL	4	Fpb
19	R/O	FL	4	Ftb
20	R/O	FL	4	Fa

Table A-13. Point Type 11, Tank Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Tag
1	R/W	AC	10	Units
2	R/W	TLP	3	Level Input
3	R/W	TLP	3	Meter Output (Pulse Input)
4	R/W	UI	2	Scan Period
5	R/O	UC	1	Alarm Code
6	R/O	UC	1	(Not used)
7	R/W	FL	4	Rate Alarm EU
8	R/W	FL	4	Strapping Value
9	R/W	FL	4	Specific Gravity
10	R/W	FL	4	Level Deadband
11	R/W	FL	4	Manual Entry
12	R/O	FL	4	Total Units Hauled
13	R/O	FL	4	Current Fluid Level
14	R/O	FL	4	Contract Hour Level
15	R/O	FL	4	Units Discharged
16	R/O	FL	4	Today's Volume
17	R/O	FL	4	Yesterday's Volume
18	R/O	FL	4	Last Scan Level
19	R/O	FL	4	Corrected Base PI

Table A-14. Point Type 12, Clock Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	UC	1	Seconds
1	R/W	UC	1	Minutes
2	R/W	UC	1	Hours
3	R/W	UC	1	Day
4	R/W	UC	1	Month
5	R/W	UC	1	Year
6	R/W	UC	1	Leap Year
7	R/W	UC	1	Day of Week
8	R/O	AC	6	Date and Time

Table A-15. Point Type 13, System Flag Parameters

Parm Number	Read-Write	DataType	Length	Description
0	R/W	UC	1	CRC Check
1	R/W	UC	1	DI/PI for ROC306/312
2	R/W	UC	1	User LCD Enable
3	R/W	UC	1	User Operator Port Enable
4	R/W	UC	1	FST/Display Clear
5	R/W	UC	1	User Com1 Enable
6	R/W	UC	1	User Com2 Enable
7	R/W	UC	1	User Program Enable
8	R/W	UC	1	RTS Operator Interface Port
9	R/W	UC	1	RTS Communications Port 1
10	R/W	UC	1	RTS Communications Port 2
11	R/W	UC	1	Clear EEPROM
12	R/W	UC	1	I/O Scan Enable
13	R/W	UC	1	Auxiliary Output 2 On
14	R/W	UC	1	Auxiliary Output 1 On
15	R/W	UC	1	Cold Hard Start
16	R/W	UC	1	Warm Start
17	R/W	UC	1	Read I/O
18	R/W	UC	1	Write to EEPROM
19	R/W	UC	1	EEPROM Write Complete

Table A-16. Point Type 14, Communication Port Parameters

Parm Number	Read-Write	DataType	Length	Description
0	R/W	AC	10	Tag Id.
1	R/W	UI	2	Baud Rate
2	R/W	UC	1	Stop Bits
3	R/W	UC	1	Data Bits
4	R/W	UC	1	Parity
5	R/O	BN	1	Status
6	R/W	BN	1	Mode
7	R/W	UC	1	Key On Delay
8	R/W	UC	1	Key Off Delay
9	R/W	UC	1	Retry Count
10	R/W	UI	2	Retry Time
11	R/O	UI	2	Alarm Pointer
12	R/O	UI	2	Recv Counter Copy
13	R/O	UI	2	Retry Counter
14	R/W	UI	2	Valid Receive Counter

Table A-17. Point Type 15, System Variables

Parm. Number	Read-Write	Data Type	Length	Description
0	R/W	UC	1	ROC Address
1	R/W	UC	1	ROC Group
2	R/W	AC	20	Station Name
3	R/O	UC	1	Active PIDs
4	R/O	UC	1	Active AGAs
5	R/O	UC	1	Active Tanks
6	R/O	UC	1	Base Database Points (History 1)
7	R/O	UC	1	RAM1 Database Points (History 2)
8	R/O	UC	1	RAM2 Database Points (History 3)
9	R/O	UC	1	Force End of Day
10	R/W	UC	1	Contract Hour
11	R/O	AC	20	Version Name
12	R/O	AC	20	Fisher ID
13	R/O	AC	20	Time Created
14	R/O	AC	12	ROM Serial Number
15	R/O	AC	20	Customer Name
16	R/O	UC	1	Maximum PIDs
17	R/O	UC	1	Maximum AGAs
18	R/O	UC	1	Maximum Tanks
19	R/O	UC	1	FSTs Active
20	R/O	UC	1	RAM Installed
21	R/O	UC	1	ROM Installed
22	R/O	FL	4	MPU Loading
23	R/O	UC	1	Utilities

Table A-18. Point Type 16, FST Register Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id.
1	R/W	FL	4	Results Register
2	R/W	FL	4	Register 1
3	R/W	FL	4	Register 2
4	R/W	FL	4	Register 3
5	R/W	FL	4	Register 4
6	R/W	FL	4	Register 5
7	R/W	FL	4	Register 6
8	R/W	FL	4	Register 7
9	R/W	FL	4	Register 8
10	R/W	FL	4	Register 9
11	R/W	FL	4	Register 10
12	R/W	UL	4	Timer 1
13	R/W	UL	4	Timer 2
14	R/W	UL	4	Timer 3
15	R/W	UL	4	Timer 4
16	R/W	AC	30	Message 1
17	R/W	AC	30	Message 2
18	R/O	AC	10	Message Data
19	R/W	UC	1	Miscellaneous
20	R/W	UC	1	Miscellaneous
21	R/W	UC	1	Miscellaneous
22	R/W	UC	1	Miscellaneous
23	R/W	UC	1	Compare Flag-SVD
24	R/W	UC	1	Run Flag
25	R/W	UI	2	Code Size
26	R/W	UI	2	Instruction Pointer
27	R/W	UI	2	Execution Delay

Table A-19. Point Type 17, Soft Point Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Point Tag Id.
1	R/W	UI	2	Integer Flag
2	R/W	FL	4	Data 1
3	R/W	FL	4	Data 2
4	R/W	FL	4	Data 3
5	R/W	FL	4	Data 4
6	R/W	FL	4	Data 5
7	R/W	FL	4	Data 6
8	R/W	FL	4	Data 7
9	R/W	FL	4	Data 8
10	R/W	FL	4	Data 9
11	R/W	FL	4	Data 10
12	R/W	FL	4	Data 11
13	R/W	FL	4	Data 12
14	R/W	FL	4	Data 13
15	R/W	FL	4	Data 14
16	R/W	FL	4	Data 15
17	R/W	FL	4	Data 16
18	R/W	FL	4	Data 17
19	R/W	FL	4	Data 18
20	R/W	FL	4	Data 19
21	R/W	FL	4	Data 20

Table A-20. Point Type 19, Database Setup Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/O	FL	4	Pointer to Tag
1	R/W	UC	1	Archive Type
2	R/W	UC	1	Point Type
3	R/W	UC	1	Point/Logical Number
4	R/W	UC	1	Parameter Number

Table A-21. Point Type 20, ROC Task Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/O	UI	2	Stack Pointer
1	R/O	UI	2	Stack Segment
2	R/O	UC	1	Priority
3	R/W	UC	1	Status
4	R/O	AC	10	Task Name
5	R/O	UI	2	Child
6	R/O	UI	2	Entry Time
7	R/O	UI	2	Exit Time

Table A-22. Point Type 40, MVS (205 Sensor) Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Sensor Tag Id
1	R/W	UC	1	Sensor Address
2	R/W	UC	1	Sensor Configuration
3	R/W	UC	1	Poll Mode
4	R/W	UC	1	Spare
5	R/O	UC	1	Sensor Status
6	R/O	UC	1	Sensor Alarms
7	R/O	FL	4	Sensor Voltage
8	R/W	FL	4	DP, Diff Reading
9	R/W	FL	4	AP, Press Reading
10	R/W	FL	4	PT, Temp Reading
11	R/W	FL	4	DP, Reverse Flow
12	R/W	FL	4	Static Press Affect
13	R/W	FL	4	DP Min Scale
14	R/W	FL	4	DP Cal Value 1
15	R/W	FL	4	DP Cal Value 2
16	R/W	FL	4	DP Cal Value 3
17	R/W	FL	4	DP Cal Value 4
18	R/W	FL	4	AP Min Scale
19	R/W	FL	4	AP Cal Value 1
20	R/W	FL	4	AP Cal Value 2
21	R/W	FL	4	AP Cal Value 3
22	R/W	FL	4	AP Cal Value 4
23	R/W	FL	4	PT Min Scale
24	R/W	FL	4	PT Cal Value 1
25	R/W	FL	4	PT Cal Value 2
26	R/W	FL	4	PT Cal Value 3
27	R/W	FL	4	PT Cal Value 4
28	R/W	UC	1	Calibrate Command
29	R/W	UC	1	Calibrate Type
30	R/W	FL	4	Set Value
31	R/O	FL	4	Manual DP
32	R/O	FL	4	Manual AP
33	R/O	FL	4	Manual PT

Table A-23. Point Type 41, AGA Run Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Tag Id
1	R/W	FL	4	Atmospheric Pressure
2	R/W	UC	1	Calc Method II
3	R/W	UC	3	Spare
4	R/W	FL	4	Pipe Reference Temp
5	R/W	UC	1	Pipe Material
6	R/W	UC	1	Factor Entry Def'n
7	R/W	FL	4	Fb Factor
8	R/W	FL	4	Fr Factor
9	R/W	FL	4	Expansion Factor
10	R/W	FL	4	Fpb Factor
11	R/W	FL	4	Ftb Factor
12	R/W	FL	4	Ftf Factor
13	R/W	FL	4	Fgr Factor
14	R/W	FL	4	Fpv Factor
15	R/W	UC	1	1 History Point
16	R/W	UC	1	Rollup
17	R/W	TLP	3	TLP
18	R/W	FL	4	Conv
19	R/W	UC	1	2 History Point
20	R/W	UC	1	Rollup
21	R/W	TLP	3	TLP
22	R/W	FL	4	Conv
23	R/W	UC	1	3 History Point
24	R/W	UC	1	Rollup
25	R/W	TLP	3	TLP
26	R/W	FL	4	Conv
27	R/W	UC	1	4 History Point
28	R/W	UC	1	Rollup
29	R/W	TLP	3	TLP
30	R/W	FL	4	Conv
31	R/W	UC	1	5 History Point
32	R/W	UC	1	Rollup
33	R/W	TLP	3	TLP
34	R/W	FL	4	Conv
35	R/W	UC	1	6 History Point
36	R/W	UC	1	Rollup
37	R/W	TLP	3	TLP
38	R/W	FL	4	Conv

Table A-23. Point Type 41, AGA Run Parameters (continued)

Parm Number	Read-Write	Data Type	Length	Description
39	R/W	UC	1	7 History Point
40	R/W	UC	1	Rollup
41	R/W	TLP	3	TLP
42	R/W	FL	4	Conv
43	R/W	UC	1	8 History Point
44	R/W	UC	1	Rollup
45	R/W	TLP	3	TLP
46	R/W	FL	4	Conv
47	R/W	UC	1	9 History Point
48	R/W	UC	1	Rollup
49	R/W	TLP	3	TLP
50	R/W	FL	4	Conv
51	R/W	UC	1	10 History Point
52	R/W	UC	1	Rollup
53	R/W	TLP	3	TLP
54	R/W	FL	4	Conv

Table A-24. Point Type 42, Extra AGA Run Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Tag Id
1	R/W	FL	4	Flow Today
2	R/W	FL	4	Flow Yesterday
3	R/W	FL	4	Flow Month
4	R/W	FL	4	Flow Prev Month
5	R/W	FL	4	Flow Accumulated
6	R/W	FL	4	Minutes Today
7	R/W	FL	4	Minutes Yesterday
8	R/W	FL	4	Minutes Month
9	R/W	FL	4	Minutes Prev Month
10	R/W	FL	4	Minutes Accumulated
11	R/W	FL	4	Energy Today
12	R/W	FL	4	Energy Yesterday
13	R/W	FL	4	Energy Month
14	R/W	FL	4	Energy Prev Month
15	R/W	FL	4	Energy Accumulated
16	R/W	FL	4	Uncorrect Today
17	R/W	FL	4	Uncorrect Yesterday
18	R/W	FL	4	Uncorrect Month
19	R/W	FL	4	Uncorrect Prev Month
20	R/W	FL	4	Uncorrect Accum
21	R/O	FL	4	Orifice Diameter @Tf
22	R/O	FL	4	Pipe Diameter @Tf
23	R/O	FL	4	Beta
24	R/O	FL	4	Velocity of Approach
25	R/O	FL	4	Coeff of Discharge
26	R/O	FL	4	Reynold's Number
27	R/O	FL	4	Upstream Pressure
28	R/O	FL	4	Molecular Weight

Table A-25. Point Type 43, User List Parameters

Parm Number	Read-Write	Data Type	Length	Description
0	R/W	AC	10	Text 1
1	R/W	AC	10	Text 2
2	R/W	AC	10	Text 3
3	R/W	AC	10	Text 4
4	R/W	AC	10	Text 5
5	R/W	AC	10	Text 6
6	R/W	AC	10	Text 7
7	R/W	AC	10	Text 8
8	R/W	AC	10	Text 9
9	R/W	AC	10	Text 10
10	R/W	AC	10	Text 11
11	R/W	AC	10	Text 12
12	R/W	AC	10	Text 13
13	R/W	AC	10	Text 14
14	R/W	AC	10	Text 15
15	R/W	AC	10	Text 16
16	R/W	TLP	3	Data 1
17	R/W	TLP	3	Data 2
18	R/W	TLP	3	Data 3
19	R/W	TLP	3	Data 4
20	R/W	TLP	3	Data 5
21	R/W	TLP	3	Data 6
22	R/W	TLP	3	Data 7
23	R/W	TLP	3	Data 8
24	R/W	TLP	3	Data 9
25	R/W	TLP	3	Data 10
26	R/W	TLP	3	Data 11
27	R/W	TLP	3	Data 12
28	R/W	TLP	3	Data 13
29	R/W	TLP	3	Data 14
30	R/W	TLP	3	Data 15
31	R/W	TLP	3	Data 16

This page intentionally blank.

APPENDIX B — PROGRAMMING EXAMPLE

The following documentation is for FST programs used by a ROC host to poll multiple field ROCs in a Modbus communications protocol environment. ***Note that this programming example demonstrates the procedures for writing an FST and should in no way be construed as a functioning program without thorough testing and study to determine its suitability in the application.***

In this example, a Fisher-Rosemount RS3 host communicates to a single ROC through an RS232 communications link using the Modbus communications protocol. This ROC then acts as a Modbus Host that brings in data from four field ROCs. The ROCs in the field use an FST to provide spontaneous report by exception (SRBX) and the host ROC uses two FSTs to control polling and SRBX response.

B.1 HOST ROC FUNCTION SEQUENCE TABLE #1

In the host ROC, FST 1 controls FST 2 and determines the type of polling to be performed. FST 1 also generates communications alarms.

Steps 0 to 4 set the default values in the FST 1 registers.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
001		SAV	@FST1,FST SEQ# 1,R10	
000		VAL	30 (UC)	
002		MND		
003		SAV	@FST1,FST SEQ# 1,R3	
004		GO	POLL	

Step 5 reads the time since midnight, and Step 6 saves the value to register 2 for reference. Step 7 initiates a poll if within 2 minutes of midnight. Steps 8 and 9 initiate a poll if the current MND minus the last MND is less than or equal to the value in register 10 (30). Step 10 is a program delay to keep MCU loading down. Steps 11 to 14 save miscellaneous flag values to registers that can be read from the FST monitor screen. Step 15 initiates a poll of all field units if the MISC1 flag equals "1" (SRBX).

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
005	START	MND		
006		SAV	@FST1,FST SEQ# 1,R2	
007		<=	2 (UC)	POLL
008		-	@FST1,FST SEQ# 1,R3	
009		>=	@FST1,FST SEQ# 1,R10	POLL
010		WT	2 (UC)	
011		VAL	@FST1,FST SEQ# 1,MISC2	
012		SAV	@FST1,FST SEQ# 1,R5	
013		VAL	@FST1,FST SEQ# 1,MISC1	
014		SAV	@FST1,FST SEQ# 1,R6	
015		==	1 (UC)	POLL

In Steps 16 through 40, current register values are compared with previous register values. If changes have occurred, all Function 16 values are sent to the field. If no changes have occurred, Step 40 sends the FST back to the Start label.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
016		VAL	@SFP11,Soft Pt 11,INT1	
017		!=	@SFP21,Soft Pt 21,INT1	SEND
018		VAL	@SFP12,Soft Pt 12,INT1	
019		!=	@SFP22,Soft Pt 22,INT1	SEND
020		VAL	@SFP13,Soft Pt 13,INT1	
021		!=	@SFP23,Soft Pt 23,INT1	SEND
022		VAL	@SFP14,Soft Pt 14,INT1	
023		!=	@SFP24,Soft Pt 24,INT1	SEND
024		VAL	@SFP1,Soft Pt 1,DATA13	
025		!=	@SFP21,Soft Pt 21,DATA13	SEND
026		VAL	@SFP2,Soft Pt 2,DATA13	
027		!=	@SFP22,Soft Pt 22,DATA13	SEND
028		VAL	@SFP1,Soft Pt 1,DATA14	
029		!=	@SFP21,Soft Pt 21,DATA14	SEND
030		VAL	@SFP2,Soft Pt 2,DATA14	
031		!=	@SFP22,Soft Pt 22,DATA14	SEND
032		VAL	@SFP1,Soft Pt 1,DATA15	
033		!=	@SFP21,Soft Pt 21,DATA15	SEND
034		VAL	@SFP2,Soft Pt 2,DATA15	
035		!=	@SFP22,Soft Pt 22,DATA15	SEND
036		VAL	@SFP1,Soft Pt 1,DATA16	
037		!=	@SFP21,Soft Pt 21,DATA16	SEND
038		VAL	@SFP2,Soft Pt 2,DATA16	
039		!=	@SFP22,Soft Pt 22,DATA16	SEND
040		GO	START	

If a value is to be sent to the field, Steps 41 through 64 will first place the new current values into softpoints for reference, and Steps 65 and 66 will place a “2” in FST2 register 1 to initiate the send sequence.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
041	SEND	VAL	@SFP11,Soft Pt 11,INT1	
042		SAV	@SFP21,Soft Pt 21,INT1	
043		VAL	@SFP12,Soft Pt 12,INT1	
044		SAV	@SFP22,Soft Pt 22,INT1	
045		VAL	@SFP13,Soft Pt 13,INT1	
046		SAV	@SFP23,Soft Pt 23,INT1	
047		VAL	@SFP14,Soft Pt 14,INT1	
048		SAV	@SFP24,Soft Pt 24,INT1	
049		VAL	@SFP1,Soft Pt 1,DATA13	
050		SAV	@SFP21,Soft Pt 21,DATA13	
051		VAL	@SFP2,Soft Pt 2,DATA13	
052		SAV	@SFP22,Soft Pt 22,DATA13	

053		VAL	@SFP1,Soft Pt 1,DATA14
054		SAV	@SFP21,Soft Pt 21,DATA14
055		VAL	@SFP2,Soft Pt 2,DATA14
056		SAV	@SFP22,Soft Pt 22,DATA14
057		VAL	@SFP1,Soft Pt 1,DATA15
058		SAV	@SFP21,Soft Pt 21,DATA15
059		VAL	@SFP2,Soft Pt 2,DATA15
060		SAV	@SFP22,Soft Pt 22,DATA15
061		VAL	@SFP1,Soft Pt 1,DATA16
062		SAV	@SFP21,Soft Pt 21,DATA16
063		VAL	@SFP2,Soft Pt 2,DATA16
064		SAV	@SFP22,Soft Pt 22,DATA16
065		VAL	2 (UC)
066		SAV	@FST2,FST SEQ# 2,R1

Steps 67 through 72 monitor the polling responses from FST2. If a ROC fails to respond for the number of retries in FST2 register 3, an event is placed in the ROC by the event routine. In Steps 69 and 70, if polling is complete, as indicated by a "0" in FST2 register 1, FST1 returns to Start.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
067	JMPS	VAL	@FST2,FST SEQ# 2,R9	R9
068		==	@FST2,FST SEQ# 2,R3	EVENT
069		VAL	@FST2,FST SEQ# 2,R1	R1
070		==	0 (UC)	START
071		WT	2 (UC)	
072		GO	JMPS	

Steps 73 and 74 place a "1" in FST2 register 1 in order to initiate a standard poll. The current MND is placed in register 3 and will be used as a reference for the next polling time with Steps 75 and 76. If a failed poll response occurs (Steps 78 and 79), the program is sent to the Event label. After polling is complete (Steps 80 and 81), the MISC1 flag is checked; if equal to "0", the program goes to Start; if equal to "1", then a "3" is placed in FST2 register 1 to initiate a poll to clear all SRBX flags (set to 0).

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
073	POLL	VAL	1 (UC)	
074		SAV	@FST2,FST SEQ# 2,R1	
075		MND		
076		SAV	@FST1,FST SEQ# 1,R3	
077	JMPU	WT	2 (UC)	
078		VAL	@FST2,FST SEQ# 2	
079		==	@FST2,FST SEQ# 2,R3	EVENT
080		VAL	@FST2,FST SEQ# 2,R1	
081		!=	0 (UC)	JMPU
082		VAL	@FST1,FST SEQ# 1,MISC1	
083		==	1 (UC)	CLEAR
084		GO	START	

Steps 85 and 86 place a value of "3" in FST2 register 1. Then Steps 88 and 89 perform poll failure check. Next, the status of the routine is verified (Steps 90 to 92) and the MISC1 flag is set to "0" (Steps 92 to 94).

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
085	CLEAR	VAL	3 (UC)	
086		SAV	@FST2,FST SEQ# 2,R1	
087	JMPC	WT	2 (UC)	
088		VAL	@FST2,FST SEQ# 2,R9	
089		==	@FST2,FST SEQ# 2,R3	EVENT
090		VAL	@FST2,FST SEQ# 2,R1	
091		!=	0 (UC)	JMPC
092		VAL	0 (UC)	
093		SAV	@FST1,FST SEQ# 1,MISC1	
094		GO	START	

Steps 95 and 96 save the FST2 register 4 value to FST1 register 8 for reference. This is the comm port Retry Count (RCOUNT parameter) value. Steps 97 through 102 write an alarm and increment the SRBX counter by "1" and save it to register 7 so that it may be read from the FST monitor screen. Steps 103 and 104 determine if polling is complete and go to the Start label if it is. If register 8 equals register 4, the FST stays in a loop to prevent constantly writing alarms to the event and alarm logs (Steps 105 and 106). FST2 register 4 points to the ROC and registers being polled. Steps 107 through 110 determine which routine the FST was performing at the time of a communications failure and cause a return to that routine.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
095	EVENT	VAL	@FST2,FST SEQ# 2,R4	
096		SAV	@FST1,FST SEQ# 1,R8	
097		MSG	COM ERROR	@FST2,FST SEQ#2,R4
098		ALM	COM ERROR	@FST2,FST SEQ#2,R4
099		VAL	@FST1,FST SEQ#1,MISC4	
100		+	1 (UC)	
101		SAV	@FST1,FST SEQ# 1,MISC4	
102		SAV	@FST1,FST SEQ# 1,R7	
103	LOOP	VAL	@FST2,FST SEQ# 2,R1	
104		==	0 (UC)	START
105		VAL	@FST1,FST SEQ# 1,R8	
106		==	@FST2,FST SEQ# 2,R4	LOOP
107		VAL	@FST2,FST SEQ# 2,R1	
108		==	3 (UC)	JMPC
109		==	1 (UC)	JMPU
110		==	2 (UC)	JMPS
111		GO	START	
112		END		
113				
114				

This ends the steps for the Host FST1. Figure B-1 shows a display of the FST1 registers screen.

FST Registers 1 of 4

Point Tag Id.	FST SEQ# 1	Timer #3	0
Result Register 1	.000000	Timer #4	0
Register #1	.0000000	Message #1	COM ERROR
Register #2	1103.000	Message #2	
Register #3	1103.000	Miscellaneous	1
Register #4	.0000000	Miscellaneous	4
Register #5	4.000000	Miscellaneous	0
Register #6	1.000000	Miscellaneous	12
Register #7	12.00000	Compare Flg-SVD	0
Register #8	6.000000	Run Flag	1
Register #9	0000000	Code Size	947
Register #10	30.00000	Instruct'n Ptr	618
Timer #1	0	Exec'n Delay	0
Timer #2	0		

Figure B-1. FST1 Registers Display

B.2 HOST ROC FUNCTION SEQUENCE TABLE #2

Steps 0 to 5 load default values in the FST2 registers. Registers 10 and 8 are polling delays in seconds, and register 3 is the number of retries.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
000		VAL	5 (UC)	
001		SAV	@FST2,FST SEQ# 2,R10	
002		VAL	10 (UC)	
003		SAV	@FST2,FST SEQ# 2,R8	
004		VAL	3 (UC)	
005		SAV	@FST2,FST SEQ# 2,R3	

Steps 6 and 7 save the current receive counter value to FST2 register 2. Steps 8 to 11 read the value of FST2 register 1 (written by FST1) and perform the type of poll requested. Step 12 inserts a 5-second wait before rechecking register 1, and Step 13 returns the FST to the Start label.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
006	START	VAL	@COM2,ROCS ,VALRCV	
007		SAV	@FST2,FST SEQ# 2,R2	
008		VAL	@FST2,FST SEQ# 2,R1	
009		==	1 (UC)	POLL2
010		==	2 (UC)	SEND2
011		==	3 (UC)	CLEAR2
012		WT	5 (UC)	
013		GO	START	

Steps 14 and15 reset the value of register 9 to 0. Steps 16 and 17 save the value of 10 to register 4 for reference, and Step 18 sends the value to the valid receive counter to direct the Modbus program to the correct line in the Host Poll configuration. Steps 19 to 21 read the value of the valid receive counter and add "2" to it. This value is used for reference in Step 30. Steps 22 to 24 initiate a poll of the ROC and wait for number of seconds specified by register 10. Steps 25 to 27 increment the value in register 9 by "1" to set the number of polls. If the poll counter value in register 9 is greater than the retry value in register 3, the program moves to the next ROC. FST1 will generate a comm failure alarm. In Steps 29 and 30, the current value of the comm receive counter is compared with the reference value in register 9. If the values are equal, the program moves to the next ROC; if they are not equal, the poll is retried.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
014	SEND2	VAL	0 (UC)	
015		SAV	@FST2,FST SEQ# 2,R9	
016	S2	VAL	10 (UC)	
017		SAV	@FST2,FST SEQ# 2,R4	
018		SAV	@COM2,ROCS ,RCOUNT	
019		VAL	@COM2,ROCS ,VALRCV	

020		+	2 (UC)	
021		SAV	@FST2,FST SEQ# 2,R2	
022		VAL	128 (UC)	
023		SAV	@COM2,ROCS ,MODE	
024		WT	@FST2,FST SEQ# 2,R10	
025		VAL	@FST2,FST SEQ# 2,R9	
026		+	1 (UC)	
027		SAV	@FST2,FST SEQ# 2,R9	
028		>	@FST2,FST SEQ# 2,R3	SEND3
029		VAL	@COM2,ROCS ,VALRCV	
030		!=	@FST2,FST SEQ# 2,R2	S2

Steps 31 through 47 work the same as the SEND2 routine above, but are used for the third ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
031	SEND3	VAL	0 (UC)	
032		SAV	@FST2,FST SEQ# 2,R9	
033	S3	VAL	22 (UC)	
034		SAV	@FST2,FST SEQ# 2,R4	
035		SAV	@COM2,ROCS ,RCOUNT	
036		VAL	@COM2,ROCS ,VALRCV	
037		+	2 (UC)	
038		SAV	@FST2,FST SEQ# 2,R2	
039		VAL	128 (UC)	
040		SAV	@COM2,ROCS ,MODE	
041		WT	@FST2,FST SEQ# 2,R10	
042		VAL	@FST2,FST SEQ# 2,R9	
043		+	1 (UC)	
044		SAV	@FST2,FST SEQ# 2,R9	
045		>	@FST2,FST SEQ# 2,R3	SEND4
046		VAL	@COM2,ROCS ,VALRCV	
047		!=	@FST2,FST SEQ# 2,R2	S3

Steps 48 through 64 work the same as the SEND2 routine, but are used for the fourth ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
048	SEND4	VAL	0 (UC)	
049		SAV	@FST2,FST SEQ# 2,R9	
050	S4	VAL	14 (UC)	
051		SAV	@FST2,FST SEQ# 2,R4	
052		SAV	@COM2,ROCS ,RCOUNT	
053		VAL	@COM2,ROCS ,VALRCV	
054		+	2 (UC)	
055		SAV	@FST2,FST SEQ# 2,R2	
056		VAL	128 (UC)	
057		SAV	@COM2,ROCS ,MODE	
058		WT	@FST2,FST SEQ# 2,R10	
059		VAL	@FST2,FST SEQ# 2,R9	
060		+	1 (UC)	
061		SAV	@FST2,FST SEQ# 2,R9	

062		>	@FST2,FST SEQ# 2,R3	SEND5
063		VAL	@COM2,ROCS ,VALRCV	
064		!=	@FST2,FST SEQ# 2,R2	S4

Steps 65 through 81 work the same as SEND2, but are used for the fifth ROC. After the last send routine, the program will proceed to poll all the ROCs for an update.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
065	SEND5	VAL	0 (UC)	
066		SAV	@FST2,FST SEQ# 2,R9	
067	S5	VAL	26 (UC)	
068		SAV	@FST2,FST SEQ# 2,R4	
069		SAV	@COM2,ROCS ,RCOUNT	
070		VAL	@COM2,ROCS ,VALRCV	
071		+	2 (UC)	
072		SAV	@FST2,FST SEQ# 2,R2	
073		VAL	128 (UC)	
074		SAV	@COM2,ROCS ,MODE	
075		WT	@FST2,FST SEQ# 2,R10	
076		VAL	@FST2,FST SEQ# 2,R9	
077		+	1 (UC)	
078		SAV	@FST2,FST SEQ# 2,R9	
079		>	@FST2,FST SEQ# 2,R3	POLL0
080		VAL	@COM2,ROCS ,VALRCV	
081		!=	@FST2,FST SEQ# 2,R2	S5

Step 82 initiates a 15-second wait to allow the I/O in the field to stabilize to new values before polling.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
082	POLL0	WT	15 (UC)	

The poll routine is the same as the send routine except that a different request is sent. Steps 83 through 99 provide the poll routine for the second ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
083	POLL2	VAL	0 (UC)	
084		SAV	@FST2,FST SEQ# 2,R9	
085	P2	VAL	0 (UC)	
086		SAV	@FST2,FST SEQ# 2,R4	
087		SAV	@COM2,ROCS ,RCOUNT	
088		VAL	@COM2,ROCS ,VALRCV	
089		+	1 (UC)	
090		SAV	@FST2,FST SEQ# 2,R2	
091		VAL	128 (UC)	
092		SAV	@COM2,ROCS ,MODE	
093		WT	@FST2,FST SEQ# 2,R8	

094		VAL	@FST2,FST SEQ# 2,R9	
095		+	1 (UC)	
096		SAV	@FST2,FST SEQ# 2,R9	
097		>	@FST2,FST SEQ# 2,R3	POLL3
098		VAL	@COM2,ROCS ,VALRCV	
099		!=	@FST2,FST SEQ# 2,R2	P2

Steps 100 through 116 work the same as the POLL2 routine, but are used for the third ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
100	POLL3	VAL	0 (UC)	
101		SAV	@FST2,FST SEQ# 2,R9	
102	P3	VAL	2 (UC)	
103		SAV	@FST2,FST SEQ# 2,R4	
104		SAV	@COM2,ROCS ,RCOUNT	
105		VAL	@COM2,ROCS ,VALRCV	
106		+	1 (UC)	
107		SAV	@FST2,FST SEQ# 2,R2	
108		VAL	128 (UC)	
109		SAV	@COM2,ROCS ,MODE	
110		WT	@FST2,FST SEQ# 2,R8	
111		VAL	@FST2,FST SEQ# 2,R9	
112		+	1 (UC)	
113		SAV	@FST2,FST SEQ# 2,R9	
114		>	@FST2,FST SEQ# 2,R3	POLL4
115		VAL	@COM2,ROCS ,VALRCV	
116		!=	@FST2,FST SEQ# 2,R2	P3

Steps 117 through 134 work the same as the POLL2 routine, but are used for the fourth ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
117	POLL4	VAL	0 (UC)	
118		SAV	@FST2,FST SEQ# 2,R9	
119	P4	VAL	4 (UC)	
120		SAV	@FST2,FST SEQ# 2,R4	
121		SAV	@COM2,ROCS ,RCOUNT	
122		VAL	@COM2,ROCS ,VALRCV	
123		+	1 (UC)	
124		SAV	@FST2,FST SEQ# 2,R2	
125		VAL	128 (UC)	
126		SAV	@COM2,ROCS ,MODE	
127		WT	@FST2,FST SEQ# 2,R8	
128		VAL	@FST2,FST SEQ# 2,R9	
129		+	1 (UC)	
130		SAV	@FST2,FST SEQ# 2,R9	
131		>	@FST2,FST SEQ# 2,R3	POLL5
132		VAL	@COM2,ROCS ,VALRCV	
133		WT	5 (UC)	
134		!=	@FST2,FST SEQ# 2,R2	P4

Steps 135 through 152 work the same as POLL2, but are used for the fifth ROC. Step 152 sends the program to the Reset routine after all the ROCs have been polled.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
135	POLL5	VAL	0 (UC)	
136		SAV	@FST2,FST SEQ# 2,R9	
137	P5	VAL	6 (UC)	
138		SAV	@FST2,FST SEQ# 2,R4	
139		SAV	@COM2,ROCS ,RCOUNT	
140		VAL	@COM2,ROCS ,VALRCV	
141		+	1 (UC)	
142		SAV	@FST2,FST SEQ# 2,R2	
143		VAL	128 (UC)	
144		SAV	@COM2,ROCS ,MODE	
145		WT	@FST2,FST SEQ# 2,R8	
146		VAL	@FST2,FST SEQ# 2,R9	
147		+	1 (UC)	
148		SAV	@FST2,FST SEQ# 2,R9	
149		>	@FST2,FST SEQ# 2,R3	RESET
150		VAL	@COM2,ROCS ,VALRCV	
151		!=	@FST2,FST SEQ# 2,R2	P5
152		GO	RESET	

The Clear routine is the same as the Send routine except that a different request is sent. Steps 153 through 169 provide the Clear routine for the second ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
153	CLEAR2	VAL	0 (UC)	
154		SAV	@FST2,FST SEQ# 2,R9	
155	CL2	VAL	31 (UC)	
156		SAV	@FST2,FST SEQ# 2,R4	
157		SAV	@COM2,ROCS ,RCOUNT	
158		VAL	@COM2,ROCS ,VALRCV	
159		+	1 (UC)	
160		SAV	@FST2,FST SEQ# 2,R2	
161		VAL	128 (UC)	
162		SAV	@COM2,ROCS ,MODE	
163		WT	@FST2,FST SEQ# 2,R10	
164		VAL	@FST2,FST SEQ# 2,R9	
165		+	1 (UC)	
166		SAV	@FST2,FST SEQ# 2,R9	
167		>	@FST2,FST SEQ# 2,R3	CLEAR3
168		VAL	@COM2,ROCS ,VALRCV	
169		!=	@FST2,FST SEQ# 2,R2	CL2

Steps 170 through 186 work the same as the CLEAR2 routine, but are used for the third ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
170	CLEAR3	VAL	0 (UC)	
171		SAV	@FST2,FST SEQ# 2,R9	
172	CL3	VAL	33 (UC)	
173		SAV	@FST2,FST SEQ# 2,R4	
174		SAV	@COM2,ROCS ,RCOUNT	
175		VAL	@COM2,ROCS ,VALRCV	
176		+	1 (UC)	
177		SAV	@FST2,FST SEQ# 2,R2	
178		VAL	128 (UC)	
179		SAV	@COM2,ROCS ,MODE	
180		WT	@FST2,FST SEQ# 2,R10	
181		VAL	@FST2,FST SEQ# 2,R9	
182		+	1 (UC)	
183		SAV	@FST2,FST SEQ# 2,R9	
184		>	@FST2,FST SEQ# 2,R3	CLEAR4
185		VAL	@COM2,ROCS ,VALRCV	
186		!=	@FST2,FST SEQ# 2,R2	CL3

Steps 187 through 203 work the same as CLEAR2, but are used for the fourth ROC.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
187	CLEAR4	VAL	0 (UC)	
188		SAV	@FST2,FST SEQ# 2,R9	
189	CL4	VAL	35 (UC)	
190		SAV	@FST2,FST SEQ# 2,R4	
191		SAV	@COM2,ROCS ,RCOUNT	
192		VAL	@COM2,ROCS ,VALRCV	
193		+	1 (UC)	
194		SAV	@FST2,FST SEQ# 2,R2	
195		VAL	128 (UC)	
196		SAV	@COM2,ROCS ,MODE	
197		WT	@FST2,FST SEQ# 2,R10	
198		VAL	@FST2,FST SEQ# 2,R9	
199		+	1 (UC)	
200		SAV	@FST2,FST SEQ# 2,R9	
201		>	@FST2,FST SEQ# 2,R3	CLEAR5
202		VAL	@COM2,ROCS ,VALRCV	
203		!=	@FST2,FST SEQ# 2,R2	CL4

Steps 204 through 221 work the same as CLEAR2, but are used for the fifth ROC. Step 221 sends the program to the Reset routine after all the ROCs have been polled.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
204	CLEAR5	VAL	0 (UC)	
205		SAV	@FST2,FST SEQ# 2,R9	
206	CL5	VAL	37 (UC)	
207		SAV	@FST2,FST SEQ# 2,R4	

208		SAV	@COM2,ROCS ,RCOUNT	
209		VAL	@COM2,ROCS ,VALRCV	
210		+	1 (UC)	
211		SAV	@FST2,FST SEQ# 2,R2	
212		VAL	128 (UC)	
213		SAV	@COM2,ROCS ,MODE	
214		WT	@FST2,FST SEQ# 2,R10	
215		VAL	@FST2,FST SEQ# 2,R9	
216		+	1 (UC)	
217		SAV	@FST2,FST SEQ# 2,R9	
218		>	@FST2,FST SEQ# 2,R3	RESET
219		VAL	@COM2,ROCS ,VALRCV	
220		!=	@FST2,FST SEQ# 2,R2	CL5
221		GO	RESET	

Steps 222 to 224 reset register 1 to a value of "0" and return the program to the start.

STEP	**LABEL**	**CMD**	**ARGUMENT1**	**ARGUMENT2**
222	RESET	VAL	0 (UC)	
223		SAV	@FST2,FST SEQ# 2,R1	
224		GO	START	
225		END		
226				

The illustrations on pages B-13 through B-21 are screen representations of various GV101 configuration parameters used in writing an FST.

FST Registers 2 of 4

Point Tag Id	FST SEQ# 2	Timer #3	0
Result Register	128.0000	Timer #4	0
Register #1	1.000000	Message #1	
Register #2	331.0000	Message #2	
Register #3	3.000000	Miscellaneous	0
Register #4	2.000000	Miscellaneous	0
Register #5	.0000000	Miscellaneous	0
Register #6	.0000000	Miscellaneous	0
Register #7	.0000000	Compare Flg-SVD	0
Register #8	10.00000	Run Flag	1
Register #9	2.000000	Code Size	1610
Register #10	10.00000	Instruct'n Ptr	1735
Timer #1	0	Exec'n Delay	0
Timer #2	0		

System Variables 1 of 1

ROC Address	1	Version Name	W68013X0082 Ver 1.61
ROC Group	2	Fisher ID	Fisher NETREX Systems
Station Name	RS3-field link	Time Created	Sep 09 13:27:04 1993
Active PID's	0	ROM Serial #	None
Active AGA's	0	Customer Name	*FISHER ROC300
Active TANK's	0	Max PID's	16
Base DB Pt's	30	Max AGA's	5
RAM1 DB Pt's	0	Max TANK's	8
RAM2 DB Pt's	0	FST Active	1
Not Used	0	RAM Installed	01101001
Contract Hour	0	ROM Installed	10000000
		MPU Loading	36.54932
		Utilities	00011101

Modbus Config COM1 1 of 1

ASCII = 0, RTU = 1:	0		
Byte Order 1=MSB 1st	1	HI Float Scale 2	.0000000
Host Enable = 1:	1	LO Float Scale 2	.0000000
Log Data 1 = Yes:	0	HI Float Scale 3	.0000000
Init Memory = 1:	0	LO Float Scale 3	.0000000
Port Switch En = 1:	0	HI Float Scale 4	.0000000
DCD=0,DI=1,SPT=2:	0	LO Float Scale 4	.0000000
Modbus Baud Rate	1200	HI Float Scale 5	.0000000
Switch Baud Rate	1200	LO Float Scale 5	.0000000
Input Data Start	100	HI Float Scale 6	.0000000
Output Data Start	300	LO Float Scale 6	.0000000
HI Integer Scale	32676	HI Float Scale 7	.0000000
LOW Integer Scale	0	LO Float Scale 7	.0000000
HI Float Scale 1	4095.000	HI Float Scale 8	.0000000
LO Float Scale 1	.0000000	LO Float Scale 8	.0000000

Modbus Funct COM1 3 of 14

TAG ID FUNCT 3A

1	Start 201	End Addr 217	Type 17	Lgl# 0	Param 2	Conv 40
2	Start 218	End Addr 234	Type 17	Lgl# 0	Param 3	Conv 40
3	Start 235	End Addr 239	Type 17	Lgl# 0	Param 4	Conv 0
4	Start 240	End Addr 240	Type 17	Lgl# 0	Param 1	Conv 0
5	Start 301	End Addr 317	Type 17	Lgl# 0	Param 5	Conv 40
6	Start 318	End Addr 334	Type 17	Lgl# 0	Param 6	Conv 40
7	Start 335	End Addr 339	Type 17	Lgl# 0	Param 7	Conv 0
8	Start 340	End Addr 340	Type 17	Lgl# 1	Param 1	Conv 0
9	Start 401	End Addr 417	Type 17	Lgl# 0	Param 8	Conv 40
10	Start 418	End Addr 434	Type 17	Lgl# 0	Param 9	Conv 40

Modbus Funct COM1 4 of 14

TAG ID FUNCT 3B

1	Start 435	End Addr 439	Type 17	Lgl# 0	Param 10	Conv 0
2	Start 440	End Addr 440	Type 17	Lgl# 2	Param 1	Conv 0
3	Start 501	End Addr 517	Type 17	Lgl# 0	Param 11	Conv 40
4	Start 518	End Addr 534	Type 17	Lgl# 0	Param 12	Conv 40
5	Start 535	End Addr 539	Type 17	Lgl# 0	Param 13	Conv 0
6	Start 540	End Addr 540	Type 17	Lgl# 3	Param 1	Conv 0
7	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Funct COM1 12 of 14

TAG ID FUNCT 16A

1	Start 241	End Addr 241	Type 17	Lgl# 10	Param 1	Conv 0
2	Start 242	End Addr 243	Type 17	Lgl# 0	Param 15	Conv 0
3	Start 244	End Addr 245	Type 16	Lgl# 128	Param 19	Conv 0
4	Start 246	End Addr 246	Type 17	Lgl# 28	Param 1	Conv 0
5	Start 341	End Addr 341	Type 17	Lgl# 11	Param 1	Conv 0
6	Start 342	End Addr 343	Type 17	Lgl# 0	Param 16	Conv 0
7	Start 344	End Addr 345	Type 16	Lgl# 128	Param 19	Conv 0
8	Start 346	End Addr 346	Type 17	Lgl# 29	Param 1	Conv 0
9	Start 441	End Addr 441	Type 17	Lgl# 12	Param 1	Conv 0
10	Start 442	End Addr 443	Type 17	Lgl# 0	Param 17	Conv 0

Modbus Funct COM1 13 of 14

TAG ID FUNCT 16B

1	Start 444	End Addr 445	Type 16	Lgl# 128	Param 19	Conv 0
2	Start 446	End Addr 446	Type 17	Lgl# 30	Param 1	Conv 0
3	Start 541	End Addr 541	Type 17	Lgl# 13	Param 1	Conv 0
4	Start 542	End Addr 543	Type 17	Lgl# 0	Param 18	Conv 0
5	Start 544	End Addr 545	Type 16	Lgl# 128	Param 19	Conv 0
6	Start 546	End Addr 546	Type 17	Lgl# 31	Param 1	Conv 0
7	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Funct COM1 6 of 14

TAG ID FUNCT 4A

1	Start 0	End Addr 27	Type 16	Lgl# 3	Param 0	Conv 0
2	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Funct COM1 10 of 14

TAG ID FUNCT 6

1	Start 0	End Addr 27	Type 16	Lgl# 3	Param 0	Conv 0
2	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Host COM1 1 of 4

HOST 09

0	RTU Addr 2	Fnc Num 3	Reg# 201	Save# 201	#Regs 40	Status 8
1	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
2	RTU Addr 3	Fnc Num 3	Reg# 301	Save# 301	#Regs 40	Status 0
3	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
4	RTU Addr 4	Fnc Num 3	Reg# 401	Save# 401	#Regs 40	Status 8
5	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
6	RTU Addr 5	Fnc Num 3	Reg# 501	Save# 501	#Regs 40	Status 8
7	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
8	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
9	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0

Modbus Host COM1 2 of 4

HOST 1019

0	RTU Addr 2	Fnc Num 16	Reg# 241	Save# 241	#Regs 3	Status 8
1	RTU Addr 2	Fnc Num 16	Reg# 246	Save# 246	#Regs 1	Status 8
2	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
3	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
4	RTU Addr 4	Fnc Num 16	Reg# 441	Save# 441	#Regs 3	Status 1
5	RTU Addr 4	Fnc Num 16	Reg# 446	Save# 446	#Regs 1	Status 1
6	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
7	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
8	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
9	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0

Modbus Host COM1 3 of 4

HOST 2029

0	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
1	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
2	RTU Addr 3	Fnc Num 16	Reg# 341	Save# 341	#Regs 3	Status 1
3	RTU Addr 3	Fnc Num 16	Reg# 346	Save# 346	#Regs 1	Status 1
4	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
5	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
6	RTU Addr 5	Fnc Num 16	Reg# 541	Save# 541	#Regs 3	Status 2
7	RTU Addr 5	Fnc Num 16	Reg# 546	Save# 546	#Regs 1	Status 3
8	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
9	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0

Modbus Host COM1 4 of 4

HOST 3039

0	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
1	RTU Addr 2	Fnc Num 16	Reg# 244	Save# 244	#Regs 1	Status 8
2	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
3	RTU Addr 3	Fnc Num 16	Reg# 344	Save# 344	#Regs 1	Status 1
4	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
5	RTU Addr 4	Fnc Num 16	Reg# 444	Save# 444	#Regs 1	Status 8
6	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
7	RTU Addr 5	Fnc Num 16	Reg# 544	Save# 544	#Regs 1	Status 8
8	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0
9	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0

Modbus Config COM2 1 of 1

ASCII = 0, RTU = 1:	1		
Byte Order 1=MSB 1st	1	HI Float Scale 2	150000.0
Host Enable = 1:	0	LO Float Scale 2	.0000000
Log Data 1 = Yes:	0	HI Float Scale 3	.0000000
Init Memory = 1:	0	LO Float Scale 3	.0000000
Port Switch En = 1:	0	HI Float Scale 4	.0000000
DCD=0,DI=1,SPT=2:	0	LO Float Scale 4	.0000000
Modbus Baud Rate	9600	HI Float Scale 5	.0000000
Switch Baud Rate	9600	LO Float Scale 5	.0000000
Input Data Start	100	HI Float Scale 6	.0000000
Output Data Start	300	LO Float Scale 6	.0000000
HI Integer Scale	32676	HI Float Scale 7	.0000000
LOW Integer Scale	0	LO Float Scale 7	.0000000
HI Float Scale 1	100000.0	HI Float Scale 8	.0000000
LO Float Scale 1	.0000000	LO Float Scale 8	.0000000

Modbus Funct COM2 3 of 14

TAG ID FUNCT 3A

1	Start 201	End Addr 217	Type 17	Lgl# 0	Param 2	Conv 39
2	Start 218	End Addr 234	Type 17	Lgl# 0	Param 3	Conv 39
3	Start 235	End Addr 239	Type 17	Lgl# 0	Param 4	Conv 1
4	Start 240	End Addr 240	Type 17	Lgl# 0	Param 1	Conv 0
5	Start 301	End Addr 317	Type 17	Lgl# 0	Param 5	Conv 39
6	Start 318	End Addr 334	Type 17	Lgl# 0	Param 6	Conv 39
7	Start 335	End Addr 339	Type 17	Lgl# 0	Param 7	Conv 1
8	Start 340	End Addr 340	Type 17	Lgl# 1	Param 1	Conv 0
9	Start 401	End Addr 417	Type 17	Lgl# 0	Param 8	Conv 39
10	Start 418	End Addr 434	Type 17	Lgl# 0	Param 9	Conv 39

Modbus Funct COM2 4 of 14

TAG ID FUNCT 3B

1	Start 435	End Addr 439	Type 17	Lgl# 0	Param 10	Conv 1
2	Start 440	End Addr 440	Type 17	Lgl# 2	Param 1	Conv 0
3	Start 501	End Addr 517	Type 17	Lgl# 0	Param 11	Conv 39
4	Start 518	End Addr 534	Type 17	Lgl# 0	Param 12	Conv 39
5	Start 535	End Addr 539	Type 17	Lgl# 0	Param 13	Conv 1
6	Start 540	End Addr 540	Type 17	Lgl# 3	Param 1	Conv 0
7	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0
8	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0
9	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0
10	Start 0	End Addr 0	Type 0	Lgl# 17	Param 0	Conv 0

Modbus Funct COM2 12 of 14

TAG ID FUNCT 16A

1	Start 241	End Addr 241	Type 17	Lgl# 10	Param 1	Conv 0
2	Start 242	End Addr 243	Type 17	Lgl# 0	Param 15	Conv 40
3	Start 244	End Addr 245	Type 16	Lgl# 128	Param 19	Conv 34
4	Start 246	End Addr 246	Type 17	Lgl# 128	Param 1	Conv 0
5	Start 341	End Addr 341	Type 17	Lgl# 11	Param 1	Conv 0
6	Start 342	End Addr 343	Type 17	Lgl# 0	Param 16	Conv 40
7	Start 344	End Addr 345	Type 16	Lgl# 128	Param 19	Conv 34
8	Start 346	End Addr 346	Type 17	Lgl# 29	Param 1	Conv 0
9	Start 441	End Addr 441	Type 17	Lgl# 12	Param 1	Conv 0
10	Start 442	End Addr 443	Type 17	Lgl# 0	Param 17	Conv 40

Modbus Funct COM2 13 of 14

TAG ID FUNCT 16B

1	Start 444	End Addr 445	Type 16	Lgl# 128	Param 19	Conv 34
2	Start 446	End Addr 446	Type 17	Lgl# 20	Param 1	Conv 0
3	Start 541	End Addr 541	Type 17	Lgl# 13	Param 1	Conv 0
4	Start 542	End Addr 543	Type 17	Lgl# 0	Param 18	Conv 40
5	Start 544	End Addr 545	Type 16	Lgl# 128	Param 19	Conv 34
6	Start 546	End Addr 546	Type 17	Lgl# 31	Param 1	Conv 0
7	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

B.3 FIELD ROC FUNCTION SEQUENCE TABLE #1

Steps 0 and 1 load default values in the FST registers upon startup. The value of 245, placed in register 10, is the time in seconds that the program will wait between SRBX attempts. This value is 240 (4 minutes) plus the ROC address times 5. This provides an offset to prevent collisions if two ROCs should perform an SRBX at the same time.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
000		VAL	245 (UC)	
001		SAV	@FST1,FST SEQ# 1,R10	

Step 2 provides a delay to reduce MCU loading. Steps 3 and 4 check the MISC1 flag to see if an SRBX is still pending. If so, a message is sent. Step 5 writes "ACK" to FST message register 1 if there is no SRBX pending. Steps 6 through 31 compare several values with reference values to see if there has been a change. If there has been a change, an SRBX alert is sent to the Host. Step 32 returns the program to the Start label.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
002	START	WT	5 (UC)	
003		VAL	@FST1,FST SEQ# 1,MISC1	
004		==	1 (UC)	SRBX
005		MSG	ACK	@FST1,FST SEQ# 1,MSG1
006		VAL	@IOA2,PT-408K,ALARM	
007		!=	@SFP1,Soft Pt 1 ,INT1	SRBX
008		VAL	@IOA4,XIT-405K ,ALARM	
009		!=	@SFP2,Soft Pt 2 ,INT1	SRBX
010		VAL	@IOA5,TT-407K,ALARM	
011		!=	@SFP3,Soft Pt 3 ,INT1	SRBX
012		VAL	@IOA6,PT-402K,ALARM	
013		!=	@SFP4,Soft Pt 4 ,INT1	SRBX
014		VAL	@IOA7,AT-409K,ALARM	
015		!=	@SFP5,Soft Pt 5 ,INT1	SRBX
016		VAL	@IOA8,AT-410K,ALARM	
017		!=	@SFP6,Soft Pt 6 ,INT1	SRBX
018		VAL	@IOA9,AT-411K,ALARM	
019		!=	@SFP7,Soft Pt 7 ,INT1	SRBX
020		VAL	@IOA10,AT-010K,ALARM	
021		!=	@SFP8,Soft Pt 8 ,INT1	SRBX
022		VAL	@IOA11,AT-011K,ALARM	
023		!=	@SFP9,Soft Pt 9 ,INT1	SRBX
024		VAL	@IOB3,ZSC-404K,STATUS	
025		!=	@SFP10,Soft Pt 10,INT1	SRBX
026		VAL	@IOB4,ZSC-406K,STATUS	
027		!=	@SFP11,Soft Pt 11,INT1	SRBX
028		VAL	@IOB5,XS-412K,STATUS	
029		!=	@SFP12,Soft Pt 12,INT1	SRBX
030		VAL	@IOA13,XY-404K,STATUS	
031		!=	@SFP13,Soft Pt 13,INT1	SRBX

032		GO	START	

Steps 33 through 70 provide the SRBX routine. Steps 33 and 34 save the value of "1" to the FST1 MISC1 parameter. This provides a flag indicating that SRBX is active. The host ROC will clear this flag after polling for information. The SRBX active indication is saved to message 1 of FST1 to display that an SRBX is pending. Steps 36 to 39 save a "0" to the Com1 Retry Count, and 128 is saved to the Mode to initiate an SRBX transmission. Steps 40 through 65 save new reference values to softpoints for future comparisons. Step 66 inserts a 10-second delay. Steps 67 and 68 force the communications port Mode to "0" to put the ROC back into the receive mode. Steps 69-70 insert a wait equal to the value in register 10 and then a minimum wait of 45 seconds. After this wait, the ROC will return to the Start label to see whether the SRBX flag is cleared and then to see if any changes have occurred.

STEP	LABEL	CMD	ARGUMENT1	ARGUMENT2
033	SRBX	VAL	1 (UC)	
034		SAV	@FST1,FST SEQ# 1,MISC1	
035		MSG	SRBX	@FST1,FST SEQ# 1,MSG1
036		VAL	0 (UC)	
037		SAV	@COM2,Comm 1,RCOUNT	
038		VAL	128 (UC)	
039		SAV	@COM2,Comm 1,MODE	
040		VAL	@IOA2,PT-408K,ALARM	
041		SAV	@SFP1,Soft Pt 1 ,INT1	
042		VAL	@IOA4,XIT-405K ,ALARM	
043		SAV	@SFP2,Soft Pt 2 ,INT1	
044		VAL	@IOA5,TT-407K,ALARM	
045		SAV	@SFP3,Soft Pt 3 ,INT1	
046		VAL	@IOA6,PT-402K,ALARM	
047		SAV	@SFP4,Soft Pt 4 ,INT1	
048		VAL	@IOA7,AT-409K,ALARM	
049		SAV	@SFP5,Soft Pt 5 ,INT1	
050		VAL	@IOA8,AT-410K,ALARM	
051		SAV	@SFP6,Soft Pt 6 ,INT1	
052		VAL	@IOA9,AT-411K,ALARM	
053		SAV	@SFP7,Soft Pt 7 ,INT1	
054		VAL	@IOA10,AT-010K,ALARM	
055		SAV	@SFP8,Soft Pt 8 ,INT1	
056		VAL	@IOA11,AT-011K,ALARM	
057		SAV	@SFP9,Soft Pt 9 ,INT1	
058		VAL	@IOB3,ZSC-404K ,STATUS	
059		SAV	@SFP10,Soft Pt 10,INT1	
060		VAL	@IOB4,ZSC-406K ,STATUS	
061		SAV	@SFP11,Soft Pt 11,INT1	
062		VAL	@IOB5,XS-412K,STATUS	
063		SAV	@SFP12,Soft Pt 12,INT1	
064		VAL	@IOA13,XY-404K,STATUS	
065		SAV	@SFP13,Soft Pt 13,INT1	
066		WT	10 (UC)	
067		VAL	0 (UC)	
068		SAV	@COM2,Comm 1,MODE	
069		WT	@FST1,FST SEQ# 1,R10	

070		WT	45	(UC)	
071		END			
072					

The following illustrations (pages B-24 through B-27) are screen representations of various GV101 configuration parameters used in writing an FST.

System Variables 1 of 1

ROC Address	4	Version Name	W68013X0082 Ver 1.61
ROC Group	2	Fisher ID	Fisher NETREX Systems
Station Name	KAHAIF #4	Time Created	Sep 09 13:27:04 1993
Active PID's	0	ROM Serial #	None
Active AGA's	1	Customer Name	*FISHER ROC300
Active TANK's	0	Max PID's	16
Base DB Pt's	10	Max AGA's	5
RAM1 DB Pt's	30	Max TANK's	8
RAM2 DB Pt's	30	FST Active	1
Not Used	0	RAM Installed	01101111
Contract Hour	0	ROM Installed	10000000
		MPU Loading	65.42194
		Utilities	00011101

Modbus Config COM1 1 of 1

ASCII = 0, RTU = 1:	0		
Byte Order 1=MSB 1st	1	HI Float Scale 2	.0000000
Host Enable = 1:	1	LO Float Scale 2	.0000000
Log Data 1 = Yes:	0	HI Float Scale 3	.0000000
Init Memory = 1:	0	LO Float Scale 3	.0000000
Port Switch En = 1:	0	HI Float Scale 4	.0000000
DCD=0,DI=1,SPT=2:	0	LO Float Scale 4	.0000000
Modbus Baud Rate	1200	HI Float Scale 5	.0000000
Switch Baud Rate	1200	LO Float Scale 5	.0000000
Input Data Start	100	HI Float Scale 6	.0000000
Output Data Start	300	LO Float Scale 6	.0000000
HI Integer Scale	32676	HI Float Scale 7	.0000000
LOW Integer Scale	0	LO Float Scale 7	.0000000
HI Float Scale 1	.0000000	HI Float Scale 8	.0000000
LO Float Scale 1	.0000000	LO Float Scale 8	.0000000

Modbus Funct COM1 3 of 14
TAG ID FUNCT 3A

1	Start 401	End Addr 417	Type 3	Lgl# 0	Param 17	Conv 0
2	Start 418	End Addr 434	Type 17	Lgl# 0	Param 1	Conv 0
3	Start 435	End Addr 436	Type 4	Lgl# 0	Param 6	Conv 0
4	Start 437	End Addr 437	Type 10	Lgl# 0	Param 3	Conv 0
5	Start 438	End Addr 438	Type 10	Lgl# 0	Param 5	Conv 0
6	Start 439	End Addr 439	Type 10	Lgl# 0	Param 7	Conv 0
7	Start 440	End Addr 440	Type 1	Lgl# 0	Param 2	Conv 37
8	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Funct COM1 6 of 14
TAG ID FUNCT 4A

1	Start 0	End Addr 27	Type 16	Lgl# 0	Param 0	Conv 0
2	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Funct COM1 10 of 14
TAG ID FUNCT 6

1	Start 0	End Addr 27	Type 16	Lgl# 0	Param 0	Conv 0
2	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Funct COM1 12 of 14
TAG ID FUNCT 16A

1	Start 441	End Addr 441	Type 2	Lgl# 0	Param 3	Conv 38
2	Start 442	End Addr 443	Type 4	Lgl# 0	Param 6	Conv 0
3	Start 444	End Addr 444	Type 16	Lgl# 0	Param 19	Conv 0
4	Start 445	End Addr 445	Type 15	Lgl# 0	Param 0	Conv 0
5	Start 446	End Addr 446	Type 17	Lgl# 31	Param 1	Conv 0
6	Start 0	End Addr 0	Type 0	Lgl# 0	Param 0	Conv 0

Modbus Host COM1 1 of 4
HOST 09

0	RTU Addr 1	Fnc Num 16	Reg# 444	Save# 444	#Regs 2	Status 8
1	RTU Addr 0	Fnc Num 0	Reg# 0	Save# 0	#Regs 0	Status 0

This page is intentionally blank.

APPENDIX C — MODEM REFERENCE

Tables C-1 through C-5 provide a reference for the AT command set used to set up a modem. The modem control configuration display allows the user to set the modem initialize and dial commands. The following paragraphs show typical commands configured for a ROC modem.

C.1 EXAMPLE MODEM COMMANDS

Initialize Command -- This command is used to initialize the modem to the desired operation parameters. A normal command for AT command set modems is as follows:

Initialize command for internal dial-up modem: ATH0E0Q0V0X0&C&S1 S0=1
Initialize command for external dial-up modem: ATH0E0Q0V0X0M1L2 S0=1

~~	=	3-second pause
+++	=	sends escape code to modem
AT*xxx*	=	disconnect and perform software reset; *xxx* depends on the type of modem, but some modems use the same command as for hang-up.

Dial Command -- This command provides the modem with a phone number to dial out. A normal command for AT command set modems is shown in the following example:

Dial Command `ATDT 9,18005151212`

where:	ATD*x*	=	dials the number 1-800-515-1212
	T	=	tone dialing
	,	=	pause (default is 2 seconds)
	9	=	access code for outside line

The comma (,) is used in this case to insert a 2-second pause after dialing a typical access code for an outside line. The delay gives time for the switchboard to connect to the outside line. The pause parameter is in the S8 register of the modem. To increase the time in 2-second increments, insert more commas. The command "ATS8 = n" where n is a number between 0 and 255 will change the delay time.

Table C-1. Modem S-Register Summary

REGISTER	RANGE	UNITS	DEFAULT	FUNCTION
S0*	0 - 255	Rings	0	Ring to Answer On
S1	0 - 255	Rings	0	Ring Count
S2	0 - 127	ASCII	43	Escape Code Character
S3	0 - 127	ASCII	13	Carriage Return Character
S4	0 - 127	ASCII	10	Line Feed Character
S5	0 -32, 127	ASCII	8	Back Space Character
S6	2 - 255	Seconds	2	Wait for Dial Tone
S7	1 - 255	Seconds	30	Wait Time for Data Carrier
S8	0 - 255	Seconds	2	Pause Time for Comma
S9	1 - 255	1/10 sec	6	Carrier Detect Response Time
S10	1 - 255	1/10 sec	7	Lost Carrier to Hang-up Delay
S11	50 - 255	Millisec	70	DTMF Dialing Speed
S12	20 - 255	1/50 sec	50	Escape Code Guard Time
S13*				Reserved
S14*				Bit Mapped Options Register
S15*			0	Reserved
S16	0 - 2			Modem SelfTest Options
S17*				Reserved
S18*	0-255	Seconds	0	Test Timer
S19				Reserved
S20				Reserved
S21*				Bit Mapped Options Register
S22*				Bit Mapped Options Register
S23*				Bit Mapped Options Register
S25*	0 - 100	Seconds	5	Delay to DTR
S26*	0 - 255	1/100 sec	1	RTS to CTS Delay Interval
S27*				Bit Mapped Options Register

* This S-Register is stored in the modem RAM upon receipt of the S= command. The RAM contents are lost when power is removed from the modem.

Table C-2. Dial Modifier Command Summary

DIAL MODIFIERS	FUNCTION
D	Dial-Originate Mode, ready to dial
P	Pulse Dial
R	Originate Call in Answer Mode
T	Touch Tone Dial
W	Wait for Dial Tone
;	Return to Idle State
@	Wait for Quiet Answer Command
!	Flash Hook
,	Pause (S8)
0-9	Dial Digits/Characters
Z	Hang up, Reset Software

Table C-3. AT Command Set Summary

BASIC COMMANDS	OPTIONS	FUNCTION
AT	UPPER CASE ONLY	Attention Code
A		Answer Commant
A/	UPPER CASE ONLY	Repeat Last Command
Bn		Communications Standard Option
C	1= ON	Squelch Transmitter
D		Dial Command
En		Off-line Character Echo Option
Fn		duplex
Hn		Switch Hook Control Option
Ln		Speaker Volume Option
Mn		Speaker Control Option
On		On-line Command
P		Pause Dial
Qn		Result Code Display Option
Sn		Select an S Register
Sn=		Write to an S register
Sn?		Read an S Register
Vn		Result Code Form Option
Xn		Result Code Set/Call Progress Option
Yn		Long Space Disconnect Option
+++		Escape Code Sequence
,	(S8) DETERMINES PT IE 9 ,	Pause
?		Returns Last Addresed S Register

Table C-4. Ampersand Command Summary

AMPERSAND COMMANDS	FUNCTION
&Cn	Data Carrier Detect Option
&Dn	Data Terminal Ready Option
&F	Load Factroy Defaults
&Gn	Guard Tone Option
&Mn	Communications Mode Option
&Pn	Make to Break Ratio Selection
&R	Clear to Send Option
&Sn	Data Set Ready Option
&Tn	Test Command Selection
&V	Display Current Profile
&Xn	Synchronous Transmit Clock Source Option

Table C-5. Result Code or Status Messages Summary

Code	Status Message	Meaning
0	OK	Command executed
1	CONNECT	Carrier detected at 300 bps
2	RING	Ring detected
3	NO CARRIER	Did not detect carrier
4	ERROR	Entry error
5	Connect 1200	Carrier detected at 1200 bps
6	No Dial Tone	Off hook, but no response after 5 seconds
7	Busy	Busy signal detected
10	Connect 2400	Carrier detected at 2400 bps

C.2 RS232 CARD TO DIAL-UP MODEM CABLE

The cable diagram shown below is for a ROC306/312/364 with an RS232 card connected to an external dial-up modem.

ROC (DB9M)	Modem (DB25M)
2	3
3	2
4	20
5	7
6	8
7	4

C.3 OPERATOR INTERFACE PORT TO DIAL-UP MODEM CABLE

The cable diagram shown below is for a ROC306/312/364 Operator Interface port connected to an external dial-up modem.

ROC (DB9M)	Modem (DB25M)
1	8
2	3
3	2
4	20
5	7
7	4

APPENDIX D — COMMUNICATION WIRING

Table D-1 shows the communications signal pin-outs for the operator interface port and the various communications cards that can be installed in a ROC300-series unit. Table D-2 shows the communications signal connections for the COM1 port and the various communications cards available for the COM2 port on a ROC407. Tables D-3 through D-6 show signal connections for ROC-to-ROC communications.

Table D-1. ROC300-Series Communications Signals

PIN	1	2	3	4	5	6	7	8	9
OPERATOR INTERFACE PORTS									
ROC306/312 OP PORT	DCD	RX	TX	DTR	COM		RTS		
ROC364 OP PORT		RX	TX	DTR	COM		RTS		+10V
COMMUNICATIONS CARDS									
EIA-232 CARD	DCD[1]	RX	TX	DTR	COM	DSR	RTS	CTS	RI[1]
EIA-422/485 CARD, 422 Usage		RX-	RX+	TX+		TX-			
EIA-422/485 CARD, 485 Usage		OUT-	OUT+						
RADIO MODEM			RXA	TXA	COM		PTT+	PTT-	
LEASED-LINE MODEM, COMM Port, 4-wire Private Line	TIP2	RING2				RING1			TIP1
LEASED-LINE MODEM, RJ-11 Port, 2-Wire Operation			TIP (RED)	RING (GRN)					
LEASED-LINE MODEM, RJ-11 Port, 4-Wire Operation		TIP2 (BLK)	TIP1 (RED)	RING 1 (GRN)	RING2 (YEL)				
DIAL-UP MODEM, RJ-11 Port			TIP (RED)	RING (GRN)					
DIAL-UP MODEM, COMM Port (output only for analyzer)	SPK	RXD	TXD	DTR	COM	RI	SHUT DOWN	+5V	DSR

[1] Signal present only on LED-equipped version of card.

Table D-2. ROC407 Communications Signals

TERMINAL	1	2	3	4	5	6	7	8	9
ROC407 COM1	RX	TX	RTS	CTS	DCD	DTR[1]	DSR	COM	
	COMMUNICATIONS CARDS (Through COM2 Only)								
EIA-232 CARD	DCD[2]	DSR	RX	RTS	TX	CTS	DTR	RI[2]	COM
EIA-422/485 CARD, 422 Usage		TX-	RX-		RX+		TX+		
EIA-422/485 CARD, 485 Usage			OUT-		OUT+				
RADIO MODEM				PTT+	RXA	PTT-	TXA		COM
LEASED-LINE MODEM, COM2 Port, 4-wire Private Line	TIP2	RING1	RING2					TIP1	
LEASED-LINE MODEM, RJ-11 Port, 2-Wire Operation			TIP (RED)	RING (GRN)					
LEASED-LINE MODEM, RJ-11 Port, 4-Wire Operation		TIP2 (BLK)	TIP1 (RED)	RING1 (GRN)	RING2 (YEL)				
DIAL-UP MODEM, RJ-11 Port			TIP (RED)	RING (GRN)					
DIAL-UP MODEM, COM2 Port (output only for analyzer)	SPK	RI	RXD	SHUT DOWN	TXD	+5V	DTR	DSR	COM

[1] Signal is permanently enabled/true.
[2] Signal present only on LED-equipped version of card.

Table D-3. ROC-to-ROC Connections for Modem Cards (4-Wire RJ-11)

ROC 1	ROC 2
Tip2 (Blk)	Tip1 (Red)
Ring1 (Grn)	Ring2 (Yel)
Tip1 (Red)	Tip2 (Blk)
Ring2 (Yel)	Ring1 (Grn)

Table D-4. ROC-to-ROC Connections for EIA-232 Serial Cards

ROC 1	ROC 2
RX	TX
TX	RX
GND	GND

Table D-5. ROC-to-ROC Connections for EIA-422 Serial Cards

ROC 1	ROC 2
RX+	TX+
TX+	RX+
RX-	TX-
TX-	RX-

Table D-6. ROC-to-ROC Connections for EIA-485 Serial Cards

ROC 1	ROC 2
OUT-	OUT-
OUT+	OUT+

This page is intentionally blank.

APPENDIX E — CONFIGURATION WORKSHEETS

The Modbus Function and Host Configuration worksheets are provided to assist the user in configuring the Modbus program. Make copies as needed.

Modbus Host Configuration Worksheet

RTU Address	Function Number	Slave Register No.	Host Register No.	Number of Registers

Modbus Function _____Configuration Worksheet

Starting Register	Ending Register	Point Type	Logical Number	Parameter Number	Conversion Code

If you have comments or questions regarding this manual, please direct them to your local sales representative or contact:

Emerson Process Management
Flow Computer Division
Marshalltown, IA 50158 U.S.A.
Houston, TX 77065 U.S.A.
Pickering, North Yorkshire UK Y018 7JA